

Image Transform



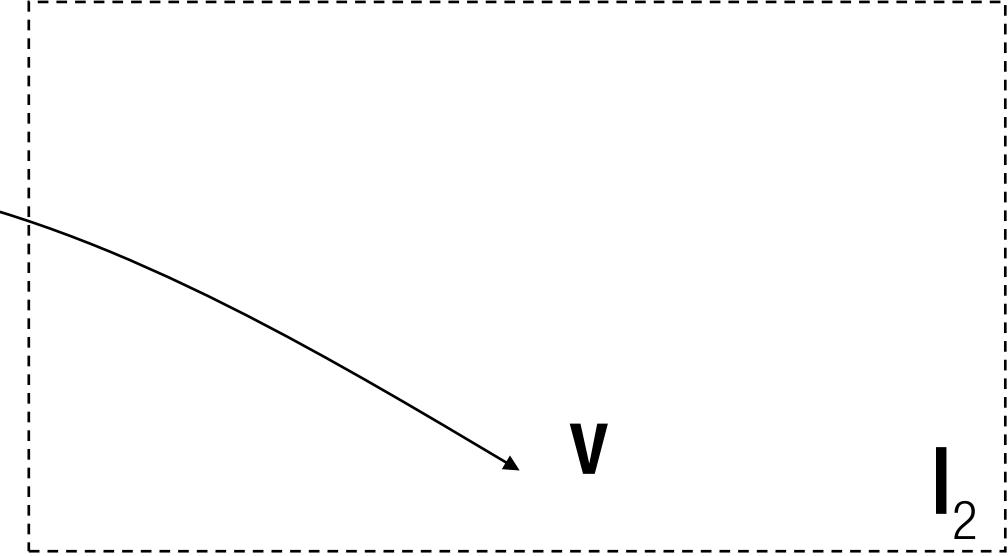
Panorama Image (Keller+Lind Hall)

Francois Vogel

Image Warping (Coordinate Transform)



I_1



v

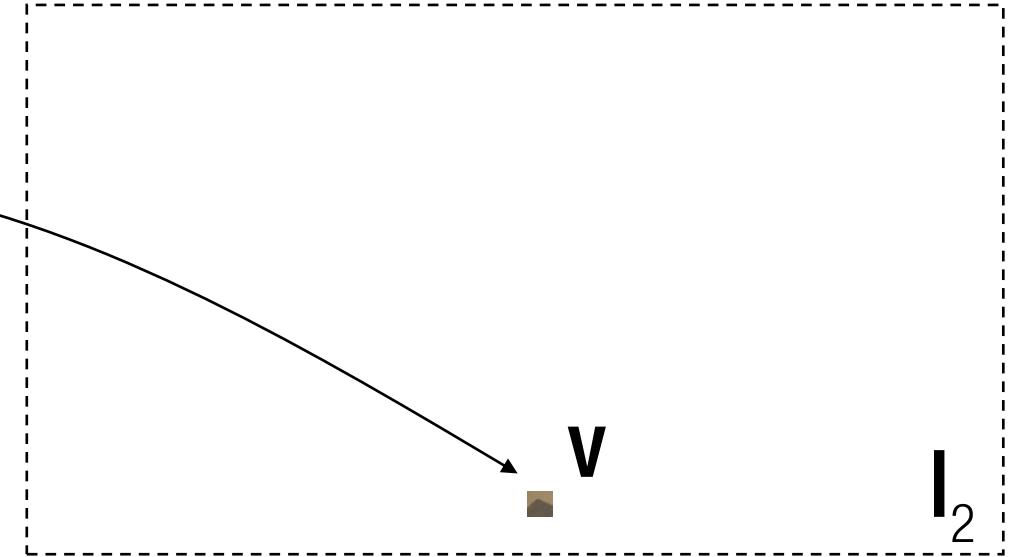
I_2

$$I_2(v) = I_1(u)$$

Image Warping (Coordinate Transform)



I₁



v

I₂

$$I_2(v) = I_1(u) \text{ : Pixel transport}$$

Hierarchy of Transformations



Euclidean (3 dof)

- Length
- Angle
- Area

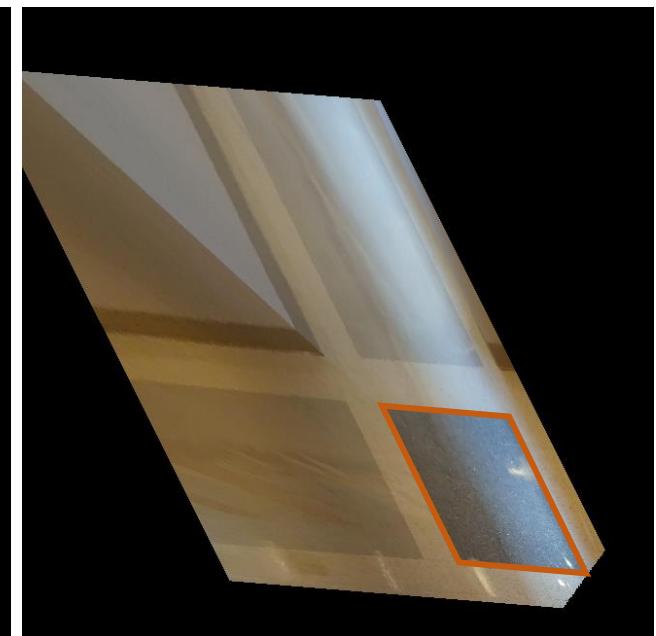
$$\begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Similarity (4 dof)

- Length ratio
- Angle

$$\begin{bmatrix} \alpha\cos\theta & -\alpha\sin\theta & t_x \\ \alpha\sin\theta & \alpha\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Affine (6 dof)

- Parallelism
- Ratio of area
- Ratio of length

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

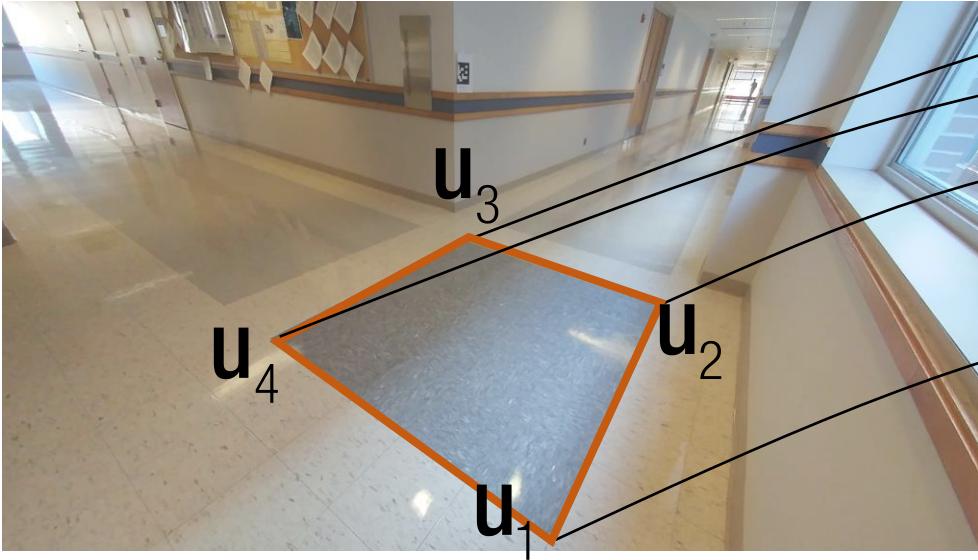


Projective (8 dof)

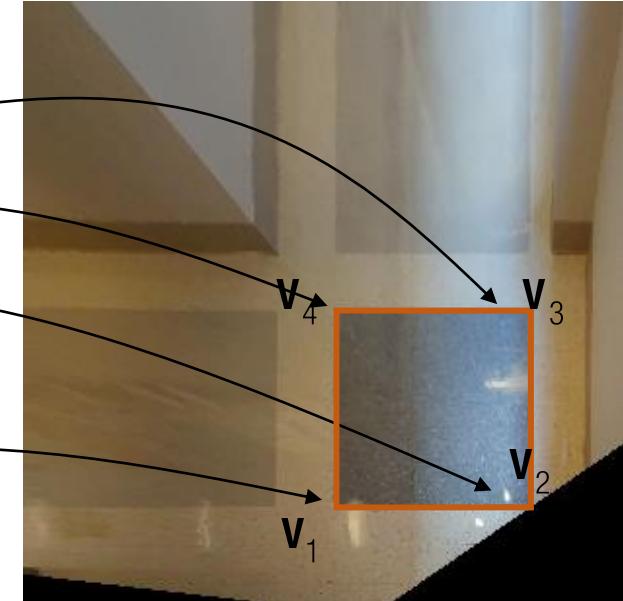
- Cross ratio
- Concurrency
- Collinearity

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$$

Fun with Homography



H



Fun with Homography

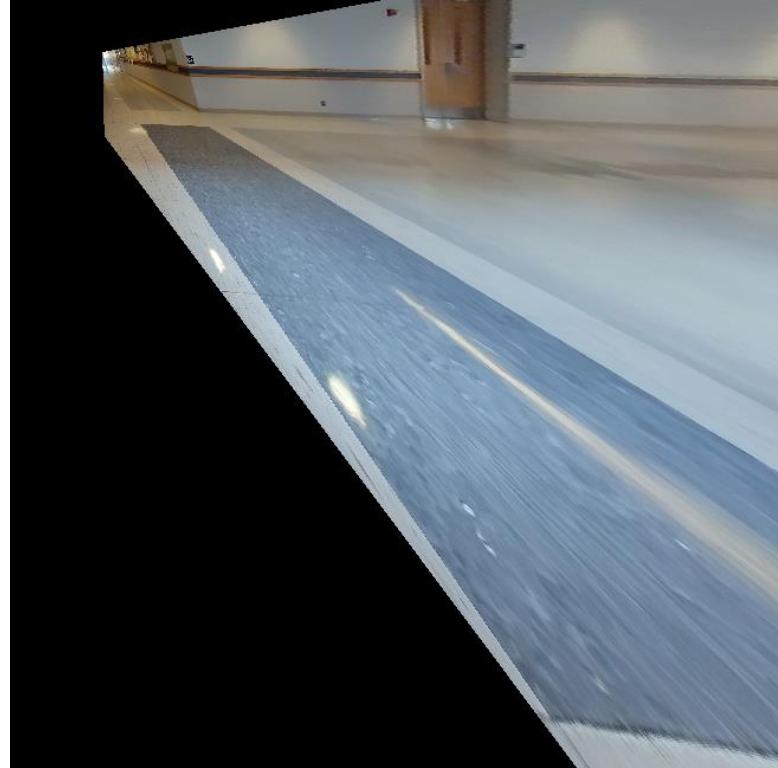


Image Inpainting



Image Inpainting



Image Transform via 3D Plane



Keller entrance left

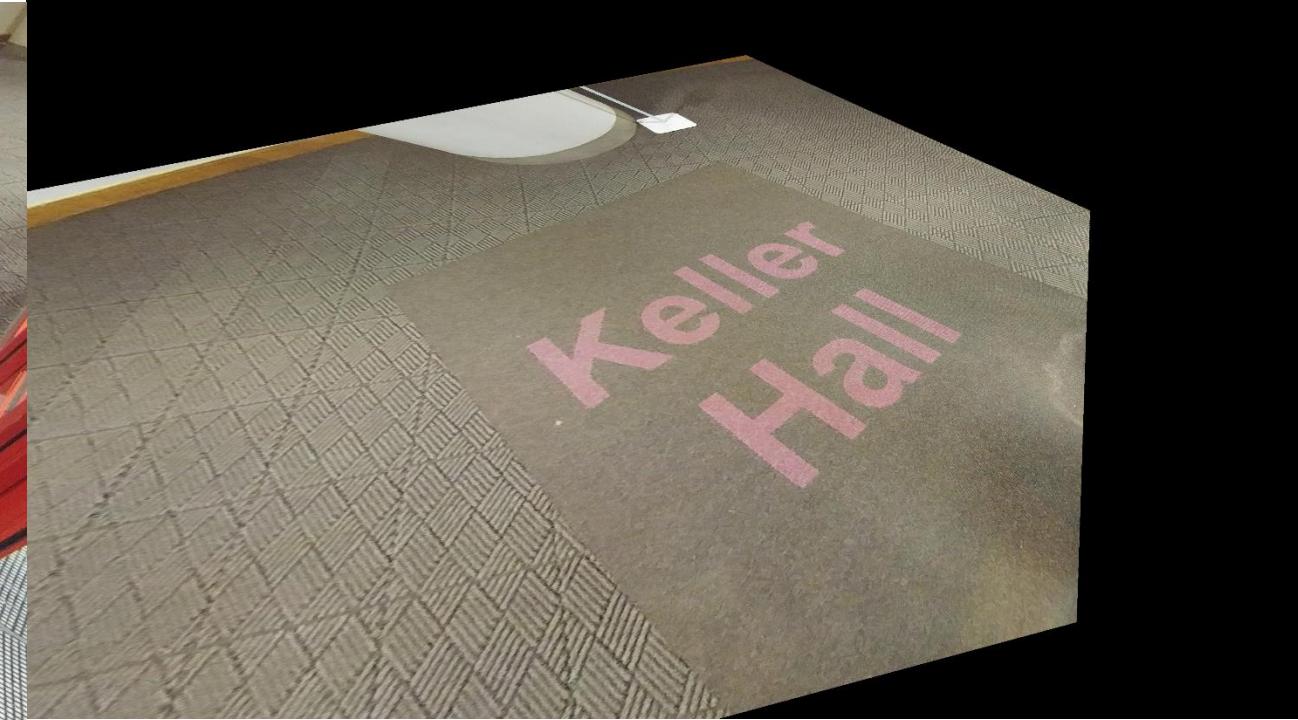


Keller entrance right

Image Transform via 3D Plane



Keller entrance left



Right image to left

Image Transform via 3D Plane



Image Transform via 3D Plane



Left image to right



Right image to left

Image Transform via 3D Plane

Keller
Hall

360 Panorama

<https://www.youtube.com/watch?v=H6SsB3JYqQg>



Image Transform by Pure 3D Rotation

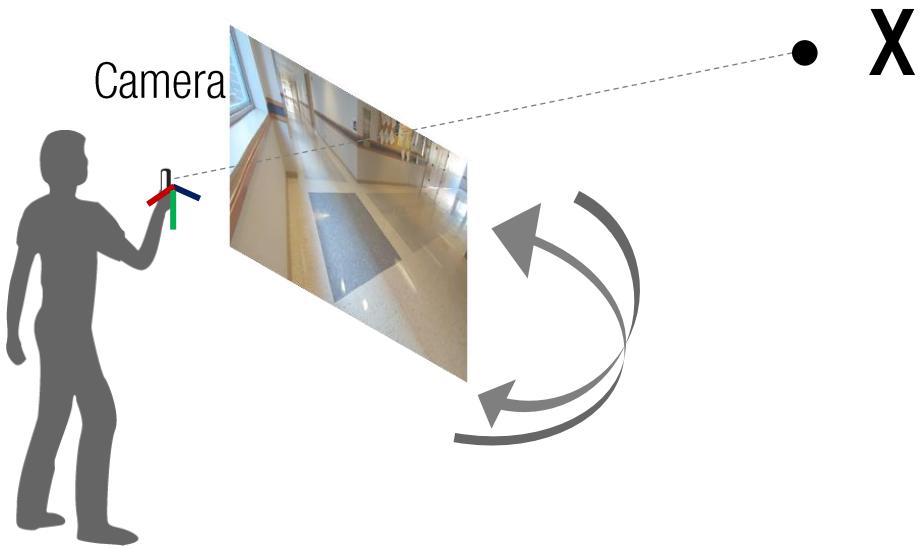


Image Transform by Pure 3D Rotation

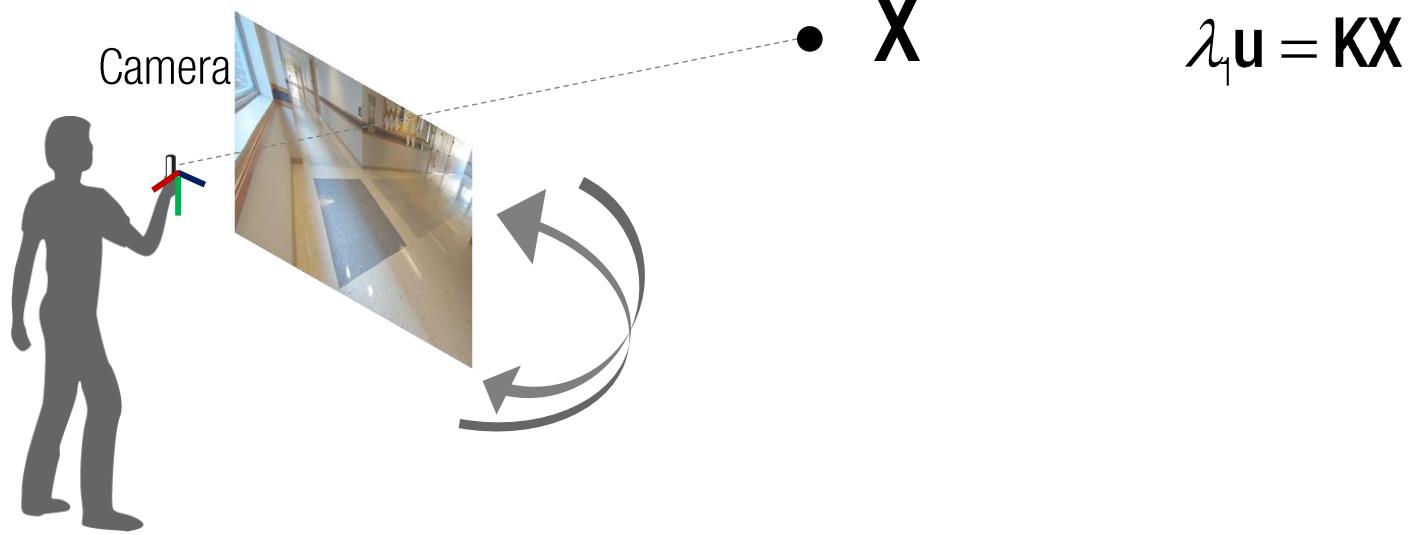
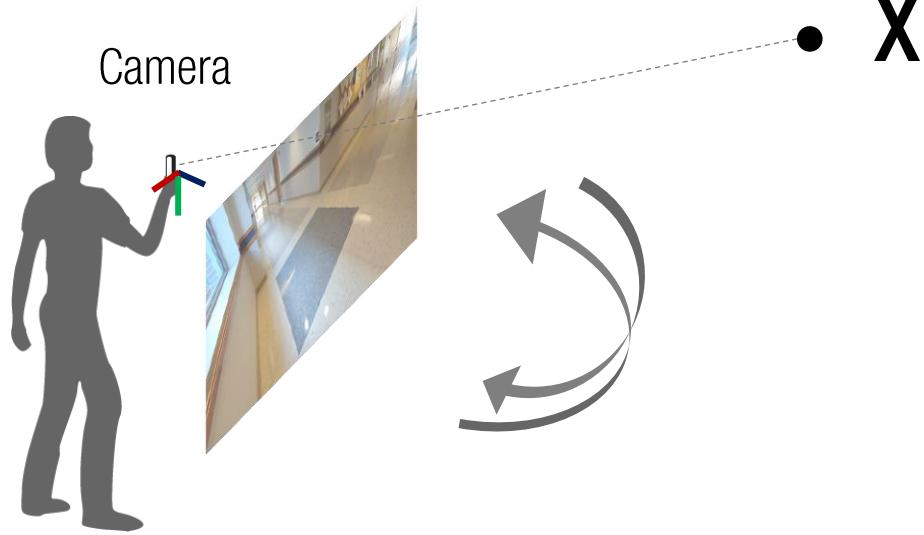


Image Transform by Pure 3D Rotation

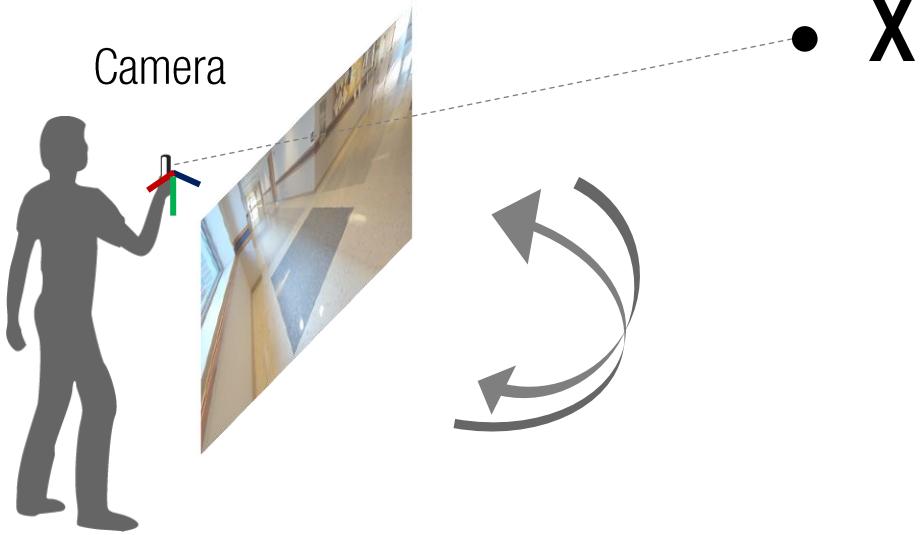


$$\lambda_1 \mathbf{u} = \mathbf{KX}$$

$$\lambda_2 \mathbf{v} = \mathbf{KRX}$$

$t = 0$

Image Transform by Pure 3D Rotation



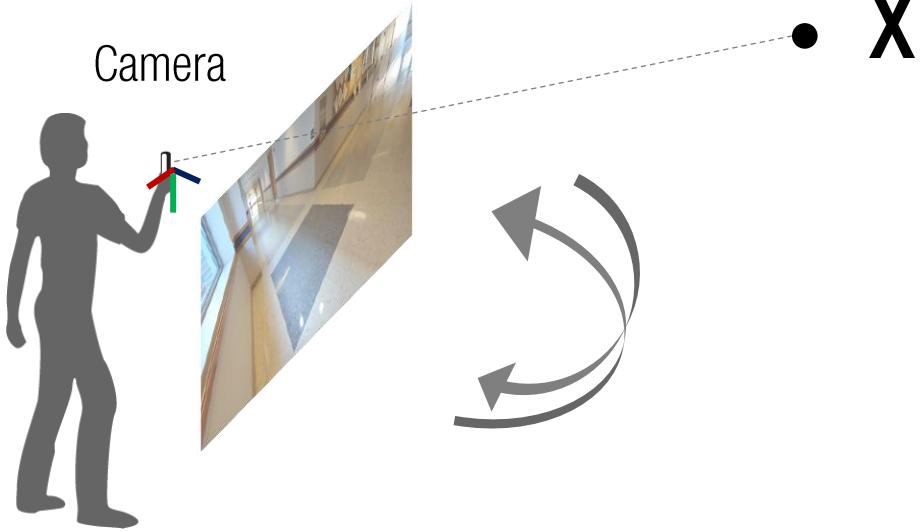
$$\lambda_1 \mathbf{u} = \mathbf{KX}$$

$$\lambda_2 \mathbf{v} = \mathbf{KRX}$$

$$t = 0$$

$$\longrightarrow \mathbf{X} = \lambda_1 \mathbf{K}^{-1} \mathbf{u} = \lambda_2 \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}$$

Image Transform by Pure 3D Rotation



$$\lambda_1 \mathbf{u} = \mathbf{KX}$$

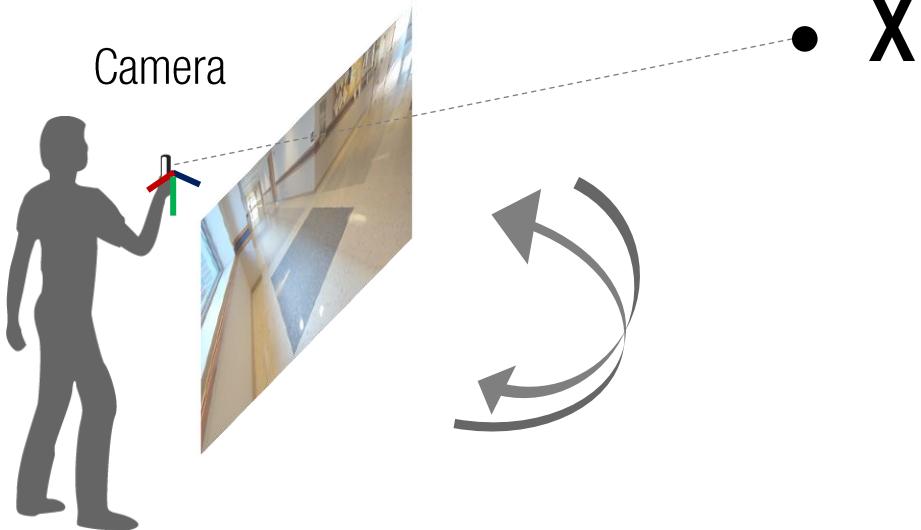
$$\lambda_2 \mathbf{v} = \mathbf{KRX}$$

$$t = 0$$

$$\longrightarrow \mathbf{X} = \lambda_1 \mathbf{K}^{-1} \mathbf{u} = \lambda_2 \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}$$

$$\longrightarrow \lambda \mathbf{v} = \mathbf{KRK}^{-1} \mathbf{u}$$

Image Transform by Pure 3D Rotation



$$\lambda_1 \mathbf{u} = \mathbf{KX}$$

$$\lambda_2 \mathbf{v} = \mathbf{KRX}$$

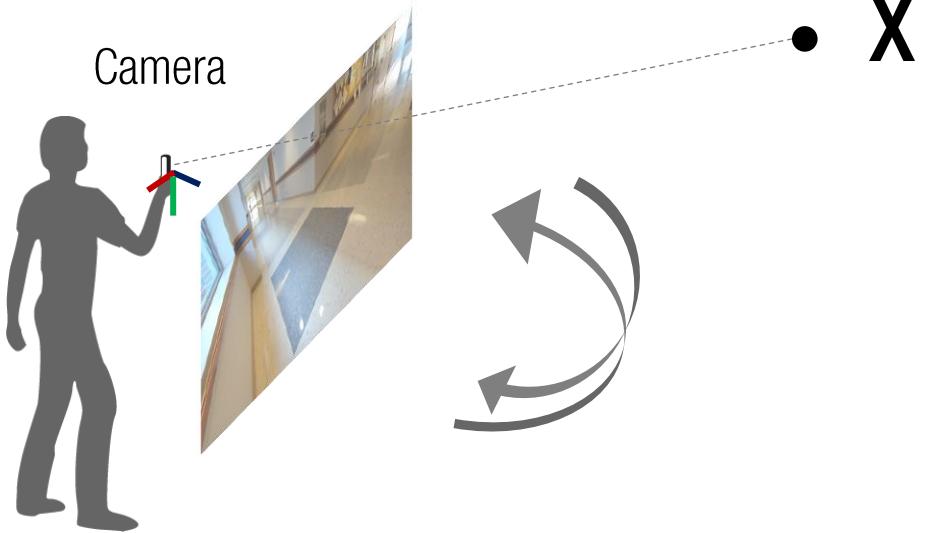
$$t = 0$$

$$\rightarrow \mathbf{X} = \lambda_1 \mathbf{K}^{-1} \mathbf{u} = \lambda_2 \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}$$

$$\rightarrow \lambda \mathbf{v} = \mathbf{KRK}^{-1} \mathbf{u}$$

$$\rightarrow \mathbf{H} = \mathbf{KRK}^{-1}$$

Image Transform by Pure 3D Rotation



$$\lambda_1 \mathbf{u} = \mathbf{K} \mathbf{X}$$

$$t = 0$$

$$\lambda_2 \mathbf{v} = \mathbf{K} \mathbf{R} \mathbf{X}$$

$$\rightarrow \mathbf{X} = \lambda_1 \mathbf{K}^{-1} \mathbf{u} = \lambda_2 \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}$$

$$\rightarrow \lambda \mathbf{v} = \mathbf{K} \mathbf{R} \mathbf{K}^{-1} \mathbf{u}$$

$$\rightarrow \mathbf{H} = \mathbf{K} \mathbf{R} \mathbf{K}^{-1}$$

$$\rightarrow \mathbf{R} = \mathbf{K}^{-1} \mathbf{H} \mathbf{K}$$

Rotation from homography

$$\lambda \mathbf{u} = \mathbf{H}\mathbf{v}$$



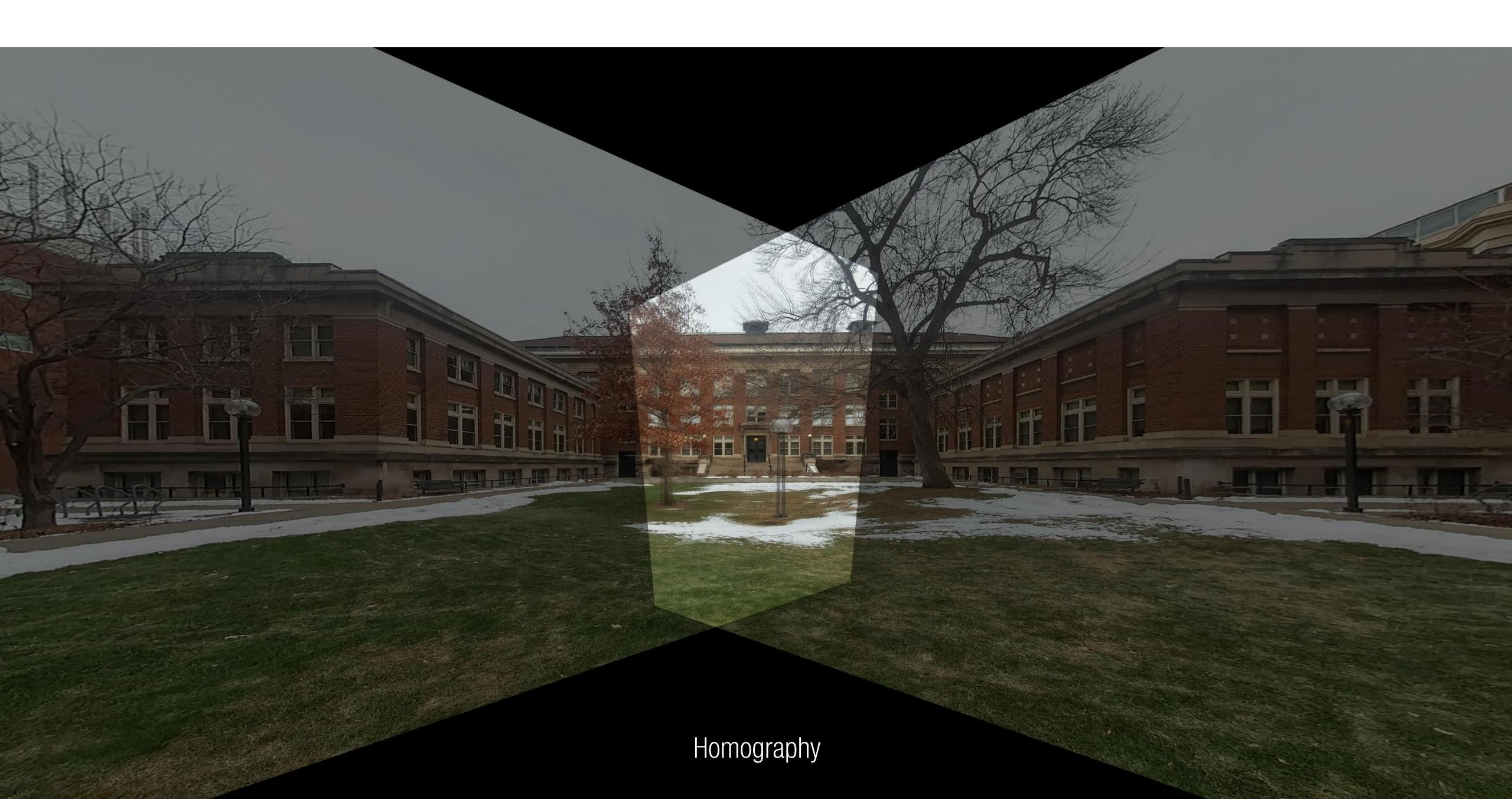
Lind Hall Left



Lind Hall Right



Euclidean Transform (Translation)



Homography

LindHallCompositeHomography.m

```
im1 = imread('lindhall_left.png');  
im2 = imread('lindhall_right.png');
```

```
f = 1224;  
px = size(im1,2)/2;  
py = size(im1,1)/2;
```

```
K = [f 0 px; 0 f py; 0 0 1];
```

```
u2 = [1275 1095; 1291 812;  
400 666; 359 1054];
```

```
u1 = [3564 1205; 3525 817;  
2624 896; 2629 1184];
```

```
H = ComputeHomography(u1,u2);  
H = H/H(3,3);  
R = inv(K) * H * K;  
detR = det(R)
```

$$H = KRK^{-1}$$

$$\longrightarrow R = K^{-1}HK$$

$$\begin{aligned} \det R = \\ 0.2059 \end{aligned}$$



LindHallCompositeHomography.m

```
im1 = imread('lindhall_left.png');  
im2 = imread('lindhall_right.png');
```

```
f = 1224;  
px = size(im1,2)/2;  
py = size(im1,1)/2;
```

```
K = [f 0 px; 0 f py; 0 0 1];
```

```
u2 = [1275 1095; 1291 812;  
400 666; 359 1054];
```

```
u1 = [3564 1205; 3525 817;  
2624 896; 2629 1184];
```

```
H = ComputeHomography(u1,u2);  
H = H/H(3,3);  
R = inv(K) * H * K;  
detR = det(R)
```

$$H = KRK^{-1}$$

$$\longrightarrow R = \lambda K^{-1}HK$$

$$\begin{aligned} \det R = \\ 0.2059 \end{aligned}$$

$$\det(\lambda R) = \lambda^3 \det(R)$$



LindHallCompositeHomography.m

```
im1 = imread('lindhall_left.png');  
im2 = imread('lindhall_right.png');
```

```
f = 1224;  
px = size(im1,2)/2;  
py = size(im1,1)/2;
```

```
K = [f 0 px; 0 f py; 0 0 1];
```

```
u2 = [1275 1095; 1291 812;  
400 666; 359 1054];
```

```
u1 = [3564 1205; 3525 817;  
2624 896; 2629 1184];
```

```
H = ComputeHomography(u1,u2);  
H = H/H(3,3);  
R = inv(K) * H * K;  
detR = det(R)
```

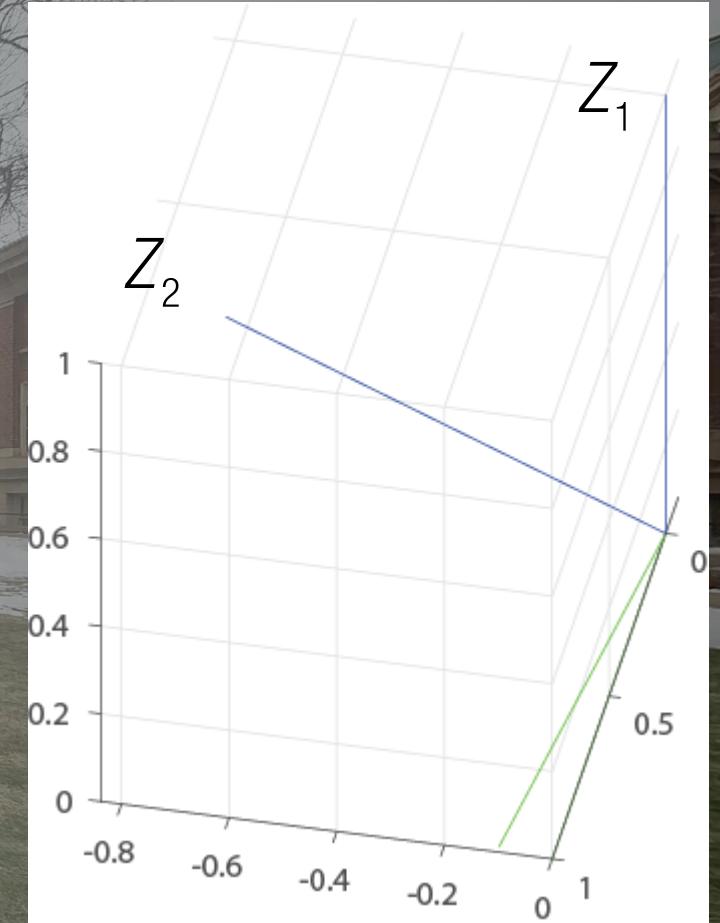
```
R = 1/detR^(1/3) * R  
det(R)
```

$$H = KRK^{-1}$$

$$\longrightarrow R = \lambda K^{-1}HK$$

$$\det R = 1.000$$

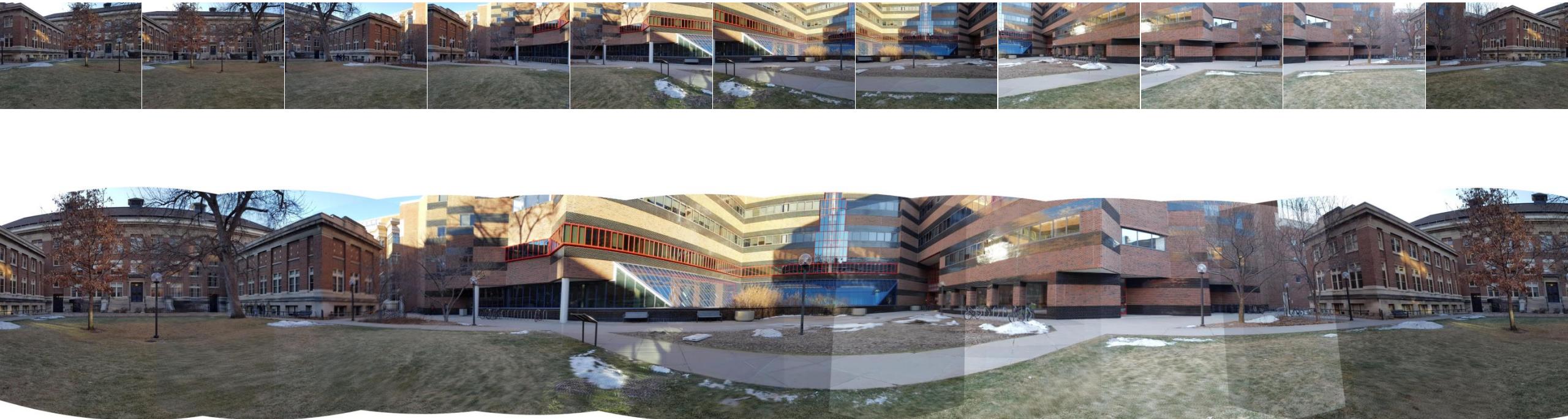
$$\det(\lambda R) = \lambda^3 \det(R)$$



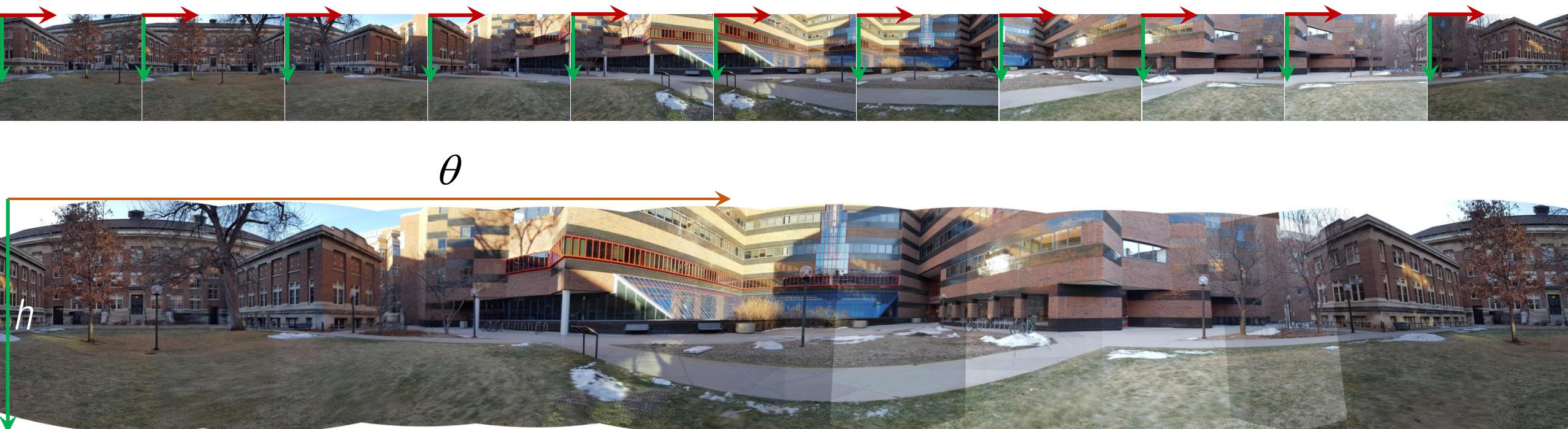
HW #2: Image Panorama (Cylindrical Projection)



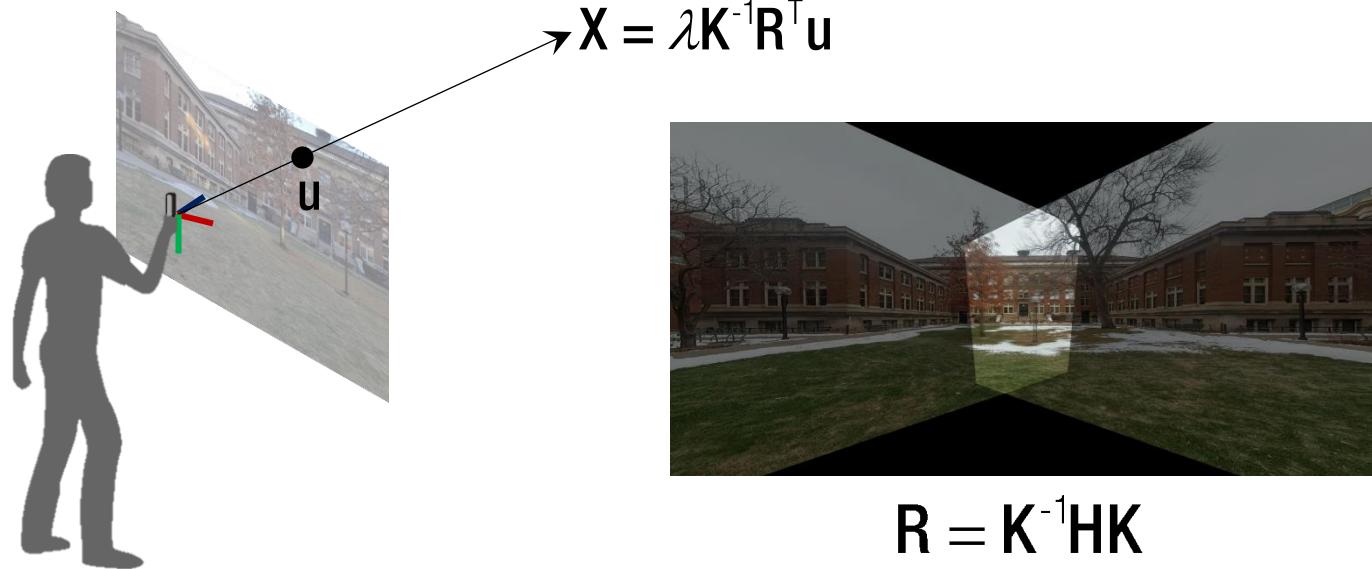
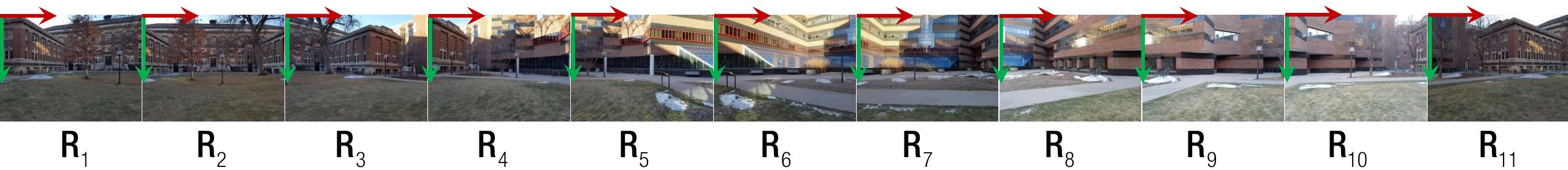
HW #2: Image Panorama (Cylindrical Projection)



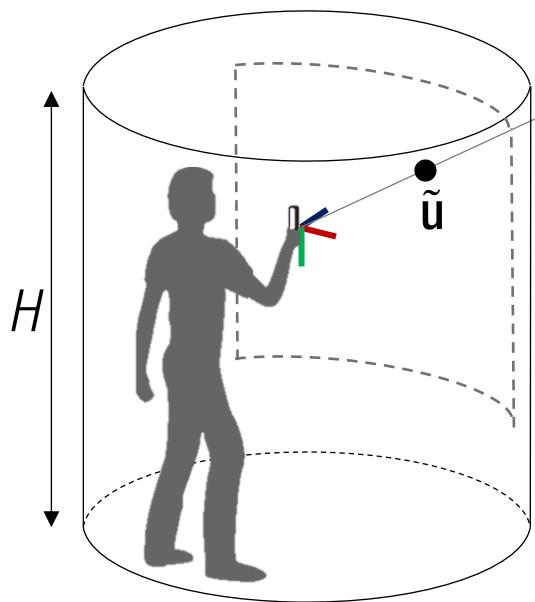
HW #2: Image Panorama (Cylindrical Projection)



HW #2: Image Panorama (Cylindrical Projection)



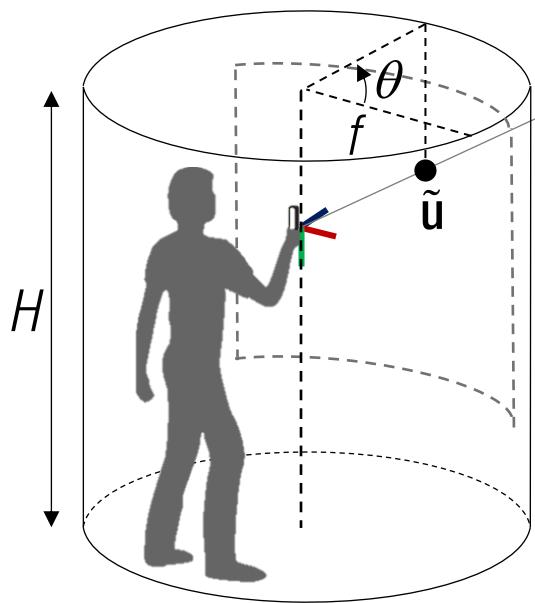
HW #2: Image Panorama (Cylindrical Projection)



$$\tilde{u} =$$

$$R = K^{-1}HK$$

HW #2: Image Panorama (Cylindrical Projection)



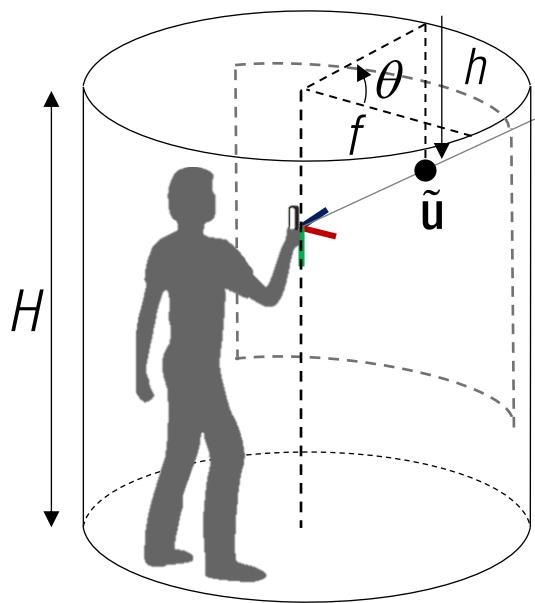
$$X = \lambda K^{-1} R^T u$$



$$R = K^{-1} H K$$

$$\tilde{u} = \begin{bmatrix} f \cos \theta \\ f \sin \theta \end{bmatrix}$$

HW #2: Image Panorama (Cylindrical Projection)



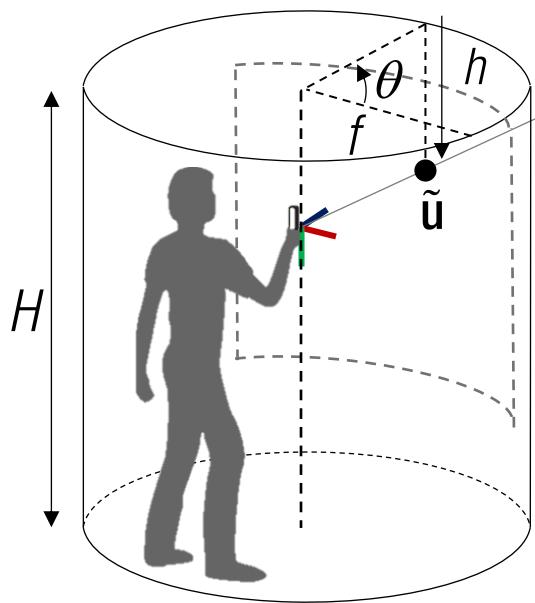
$$X = \lambda K^{-1} R^T \tilde{u}$$



$$R = K^{-1} H K$$

$$\tilde{u} = \begin{bmatrix} f \cos \theta \\ h - \frac{H}{2} \\ f \sin \theta \end{bmatrix}$$

HW #2: Image Panorama (Cylindrical Projection)



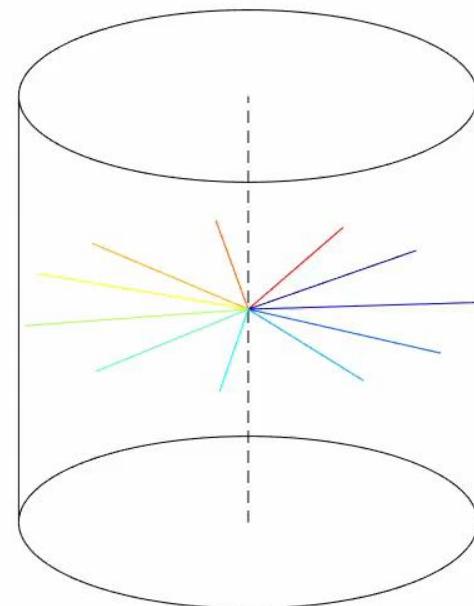
$$X = \lambda K^{-1} R^T u$$



$$R = K^{-1} H K$$

$$\tilde{u} = \begin{bmatrix} f \cos \theta \\ h - \frac{H}{2} \\ f \sin \theta \end{bmatrix} = \lambda K^{-1} R^T u$$

HW #2: Image Panorama (Cylindrical Projection)



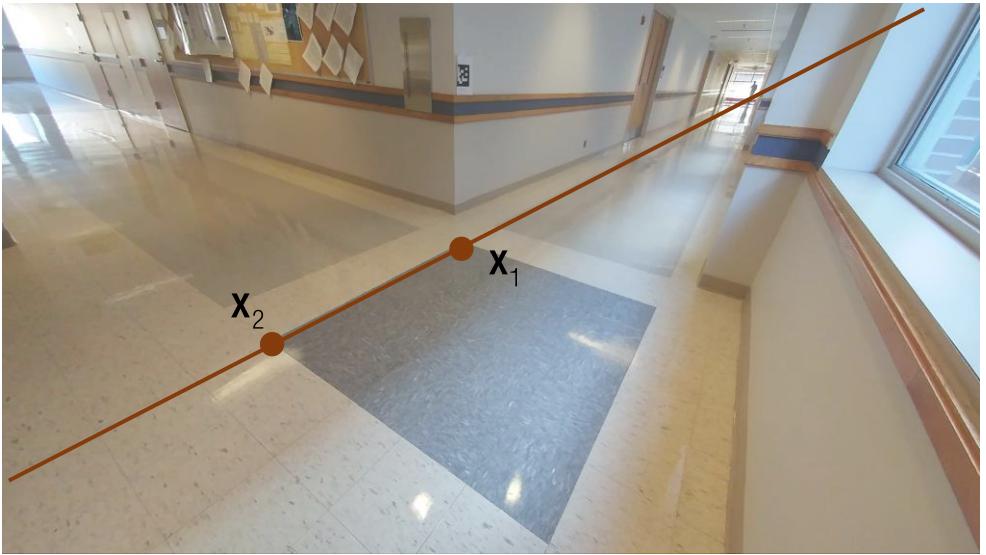


How to compute homography?

Linear Parameter Estimation



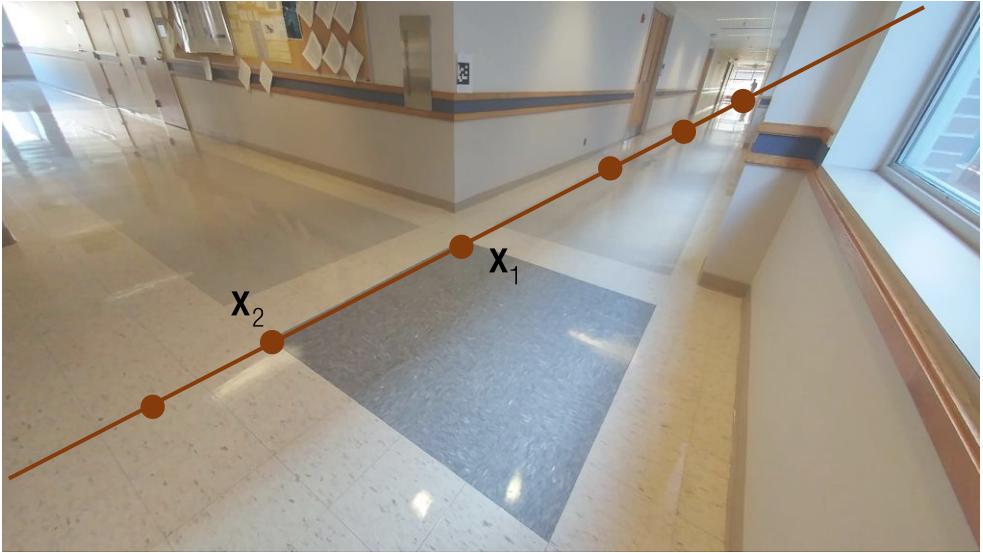
Point-Point in Image



$$\begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \end{bmatrix} \mathbf{l} = \mathbf{0}$$

$$\frac{\begin{array}{c|c} \mathbf{A} & \mathbf{l} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \hline 3 \times 2 \end{array}}{\rightarrow \mathbf{l} = \text{null}\left(\begin{array}{c|c} \mathbf{A} & \mathbf{l} \end{array}\right)} \quad \text{or} \quad \mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$$

Point-Point in Image



$$\begin{array}{c|c} \text{A} & \begin{matrix} 0 \\ 0 \end{matrix} \end{array} \rightarrow \begin{matrix} \text{I} = \text{null} \left(\begin{array}{c} \text{A} \\ \vdots \end{array} \right) \end{matrix} \quad \text{or} \quad \text{I} = \mathbf{x}_1 \times \mathbf{x}_2$$

Line Fitting

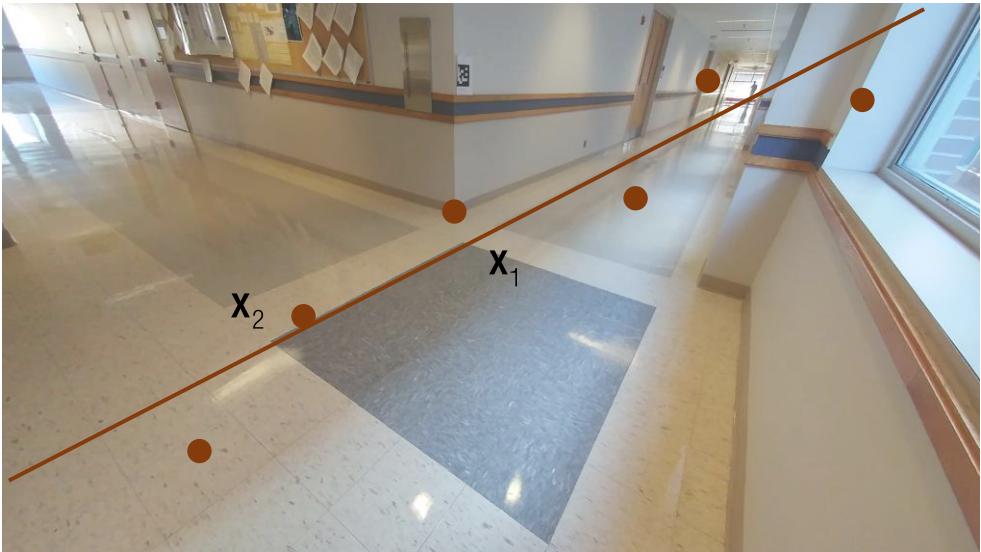


$$\begin{matrix} A \\ \hline \end{matrix} \begin{matrix} | \\ \hline \end{matrix} = \begin{matrix} 0 \\ 0 \\ \hline \end{matrix} \rightarrow ?$$

Line Fitting

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

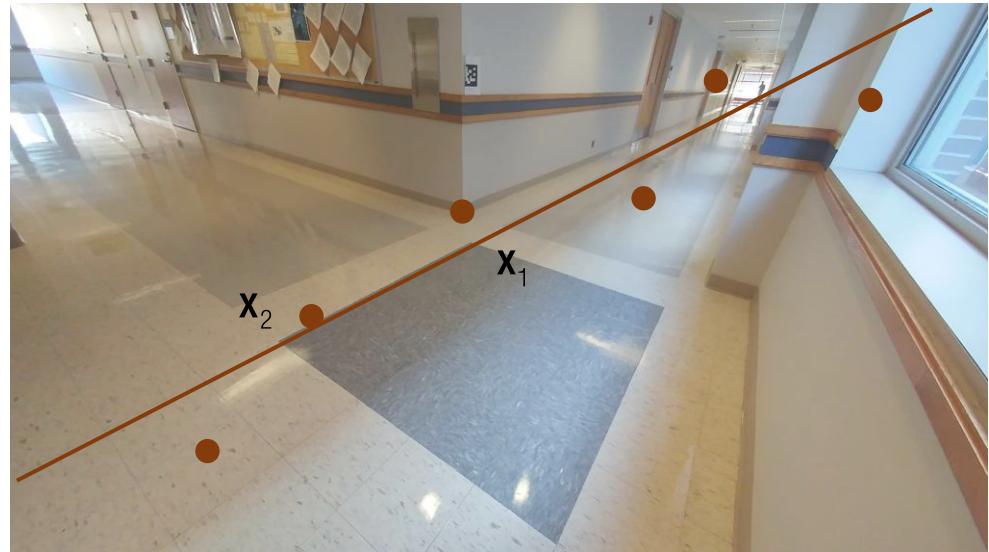
Find the best line: (a, b, c)



Line Fitting

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c)



$$\rightarrow au_1 + bv_1 + c \approx 0$$

$$au_2 + bv_2 + c \approx 0$$

⋮

$$au_n + bv_n + c \approx 0$$

Line Fitting

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \longrightarrow (\underline{m}, \underline{d})$

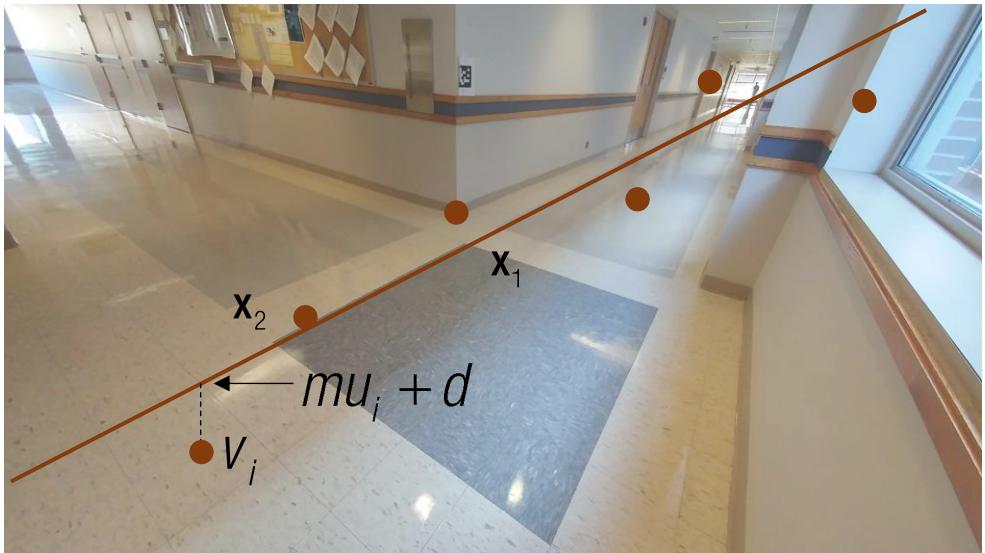
slope y-intercept

$$\longrightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$



Line Fitting

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \longrightarrow (\underline{m}, \underline{d})$

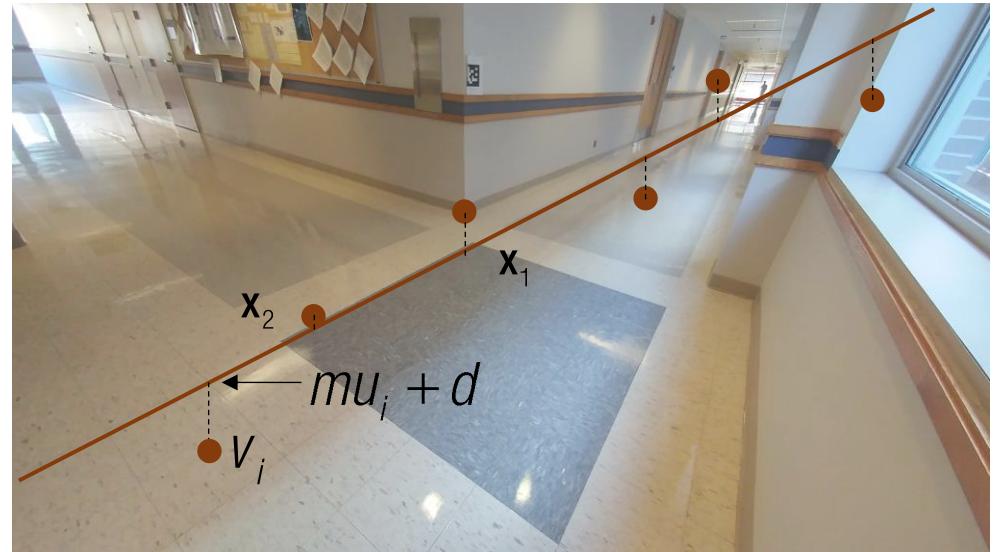
slope y-intercept

$$\longrightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$



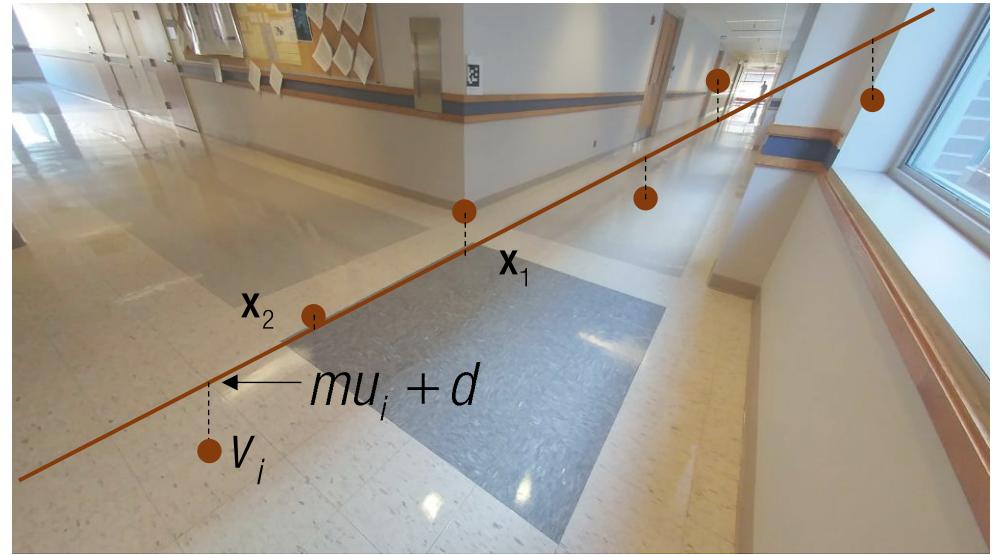
$$\text{Error: } e_i = v_i - (mu_i + d)$$

$$e_1 = v_1 - (mu_1 + d)$$

$$e_2 = v_2 - (mu_2 + d)$$

$$e_n = v_n - (mu_n + d)$$

Line Fitting



$$e_1 = v_1 - (mu_1 + d)$$

$$e_2 = v_2 - (mu_2 + d)$$

$$e_n = v_n - (mu_n + d)$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \rightarrow (\underline{m}, \underline{d})$

slope y-intercept

$$\rightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

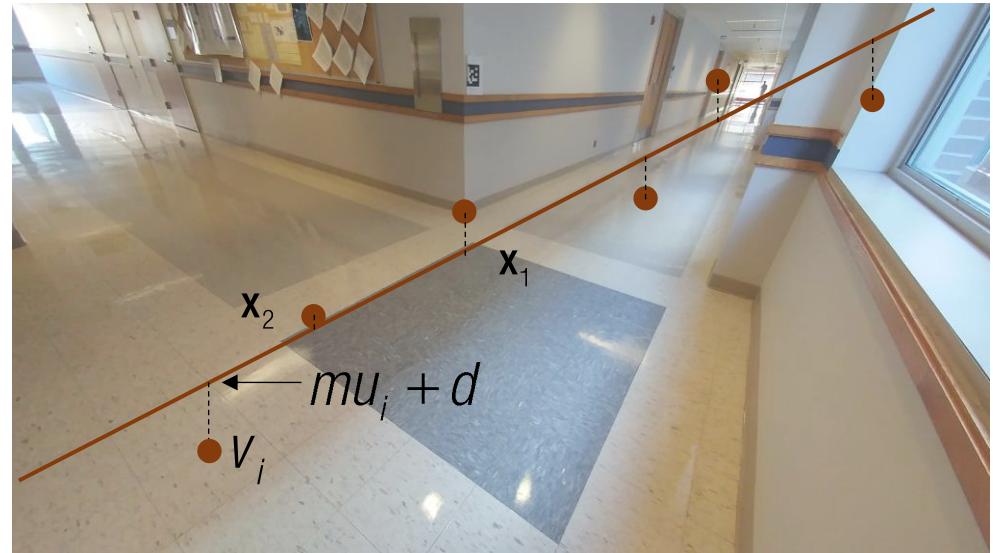
$$v_n \approx mu_n + d$$

Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Unknowns:

Number of eq.:

Line Fitting



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \longrightarrow (\underline{m}, \underline{d})$

slope y-intercept

$$\longrightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$e_1 = v_1 - (mu_1 + d)$$

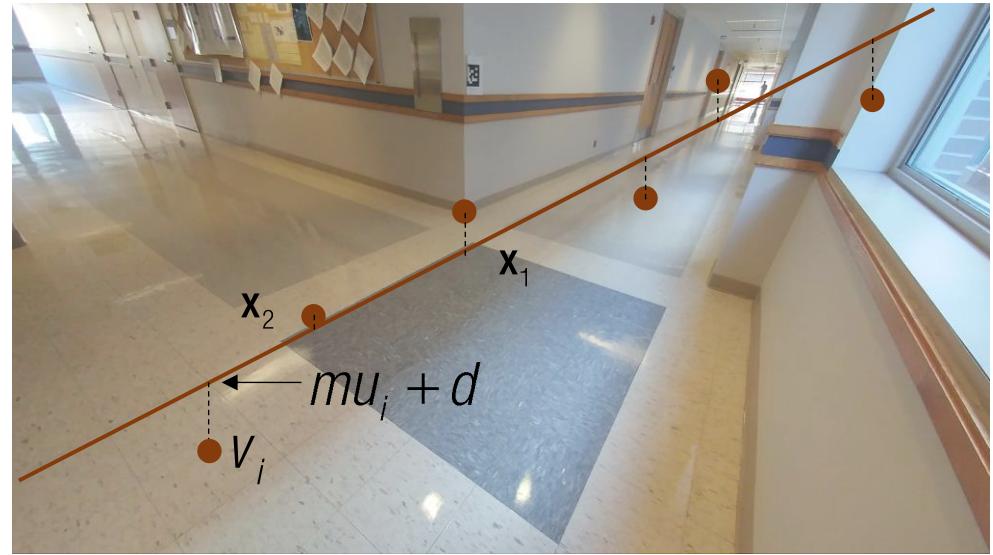
$$e_2 = v_2 - (mu_2 + d)$$

$$e_n = v_n - (mu_n + d)$$

Unknowns: m, d

Number of eq.:

Line Fitting



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \longrightarrow (\underline{m}, \underline{d})$

slope y-intercept

$$\longrightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

$$\text{Total error: } E = \sum_{i=1}^n (v_i - (mu_i + d))^2$$

$$e_1 = v_1 - (mu_1 + d)$$

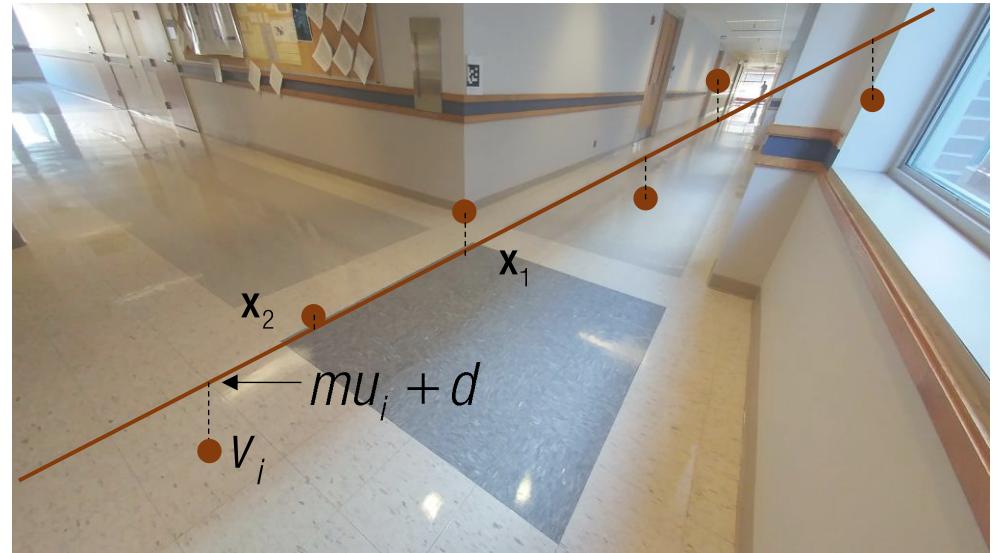
$$e_2 = v_2 - (mu_2 + d)$$

$$e_n = v_n - (mu_n + d)$$

Unknowns: m, d

Number of eq.: n

Line Fitting



$$e_1 = v_1 - (mu_1 + d)$$

$$e_2 = v_2 - (mu_2 + d)$$

$$e_n = v_n - (mu_n + d)$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \rightarrow (\underline{m}, \underline{d})$

slope y-intercept

$$\rightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

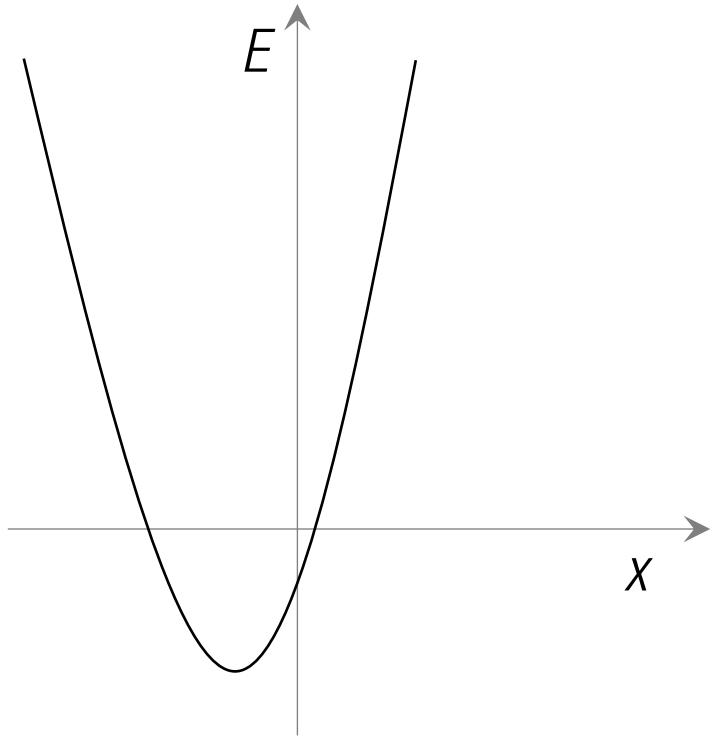
Unknowns: m, d

Number of eq.: n

$$\underset{m,d}{\text{minimize}} \sum_{i=1}^n (v_i - (mu_i + d))^2$$

How to minimize?

Line Fitting



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \rightarrow (\underline{m}, \underline{d})$
slope y-intercept

$$\rightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

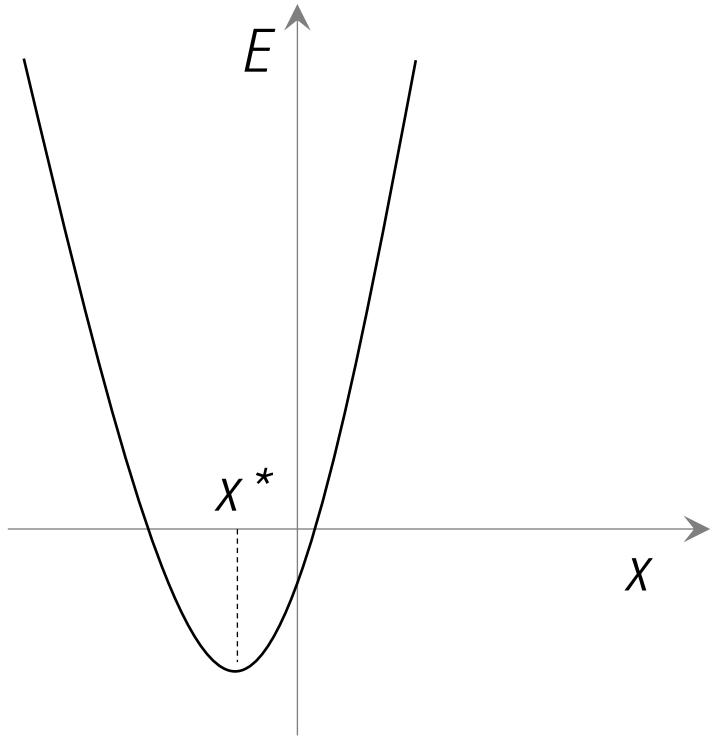
Unknowns: m, d

Number of eq.: n

$$\underset{m, d}{\text{minimize}} \sum_{i=1}^n (v_i - (mu_i + d))^2$$

How to minimize?

Line Fitting



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \rightarrow (\underline{m}, \underline{d})$
slope y-intercept

$$\rightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

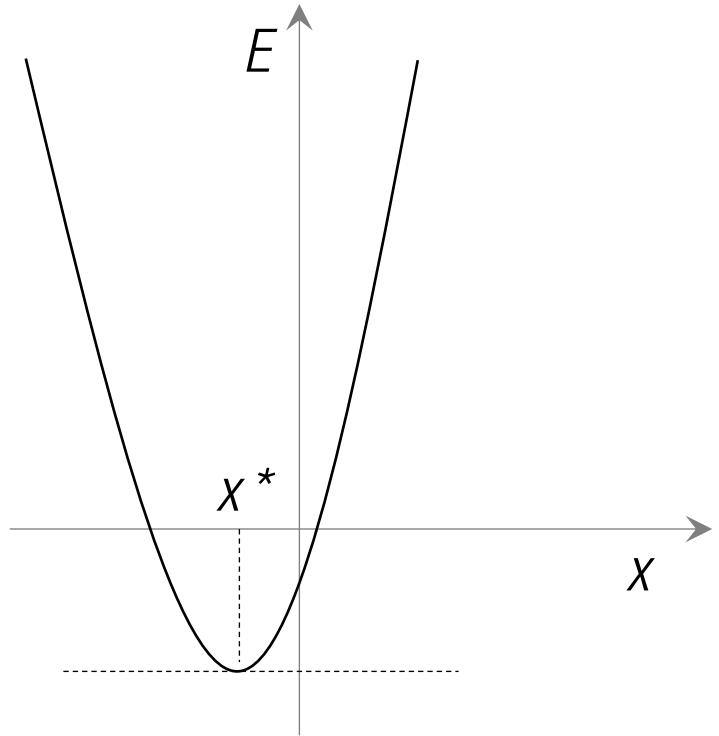
Unknowns: m, d

Number of eq.: n

$$\underset{m, d}{\text{minimize}} \sum_{i=1}^n (v_i - (mu_i + d))^2$$

How to minimize?

Line Fitting



$$\frac{dE}{dx} \Big|_{x=x^*} = 0$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \rightarrow (\underline{m}, \underline{d})$

slope y-intercept

$$\rightarrow v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

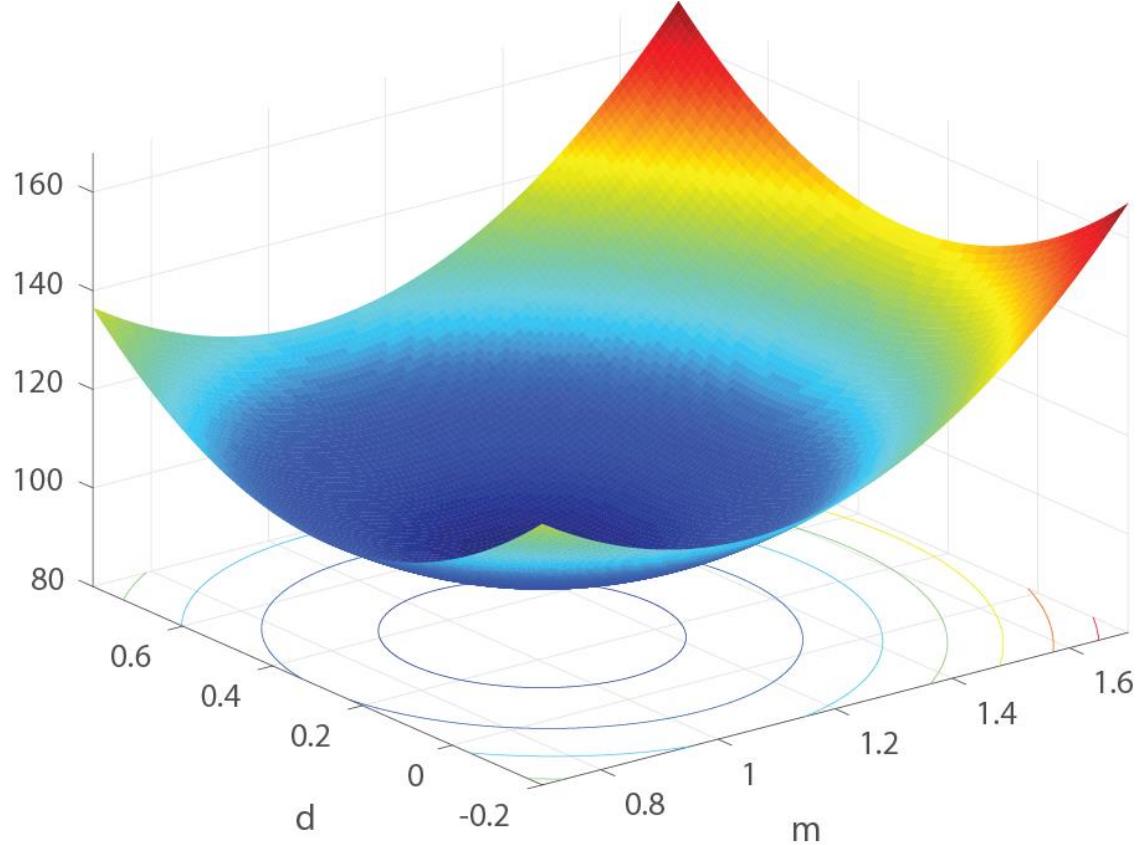
Unknowns: m, d

Number of eq.: n

$$\underset{m,d}{\text{minimize}} \sum_{i=1}^n (v_i - (mu_i + d))^2$$

How to minimize?

Line Fitting



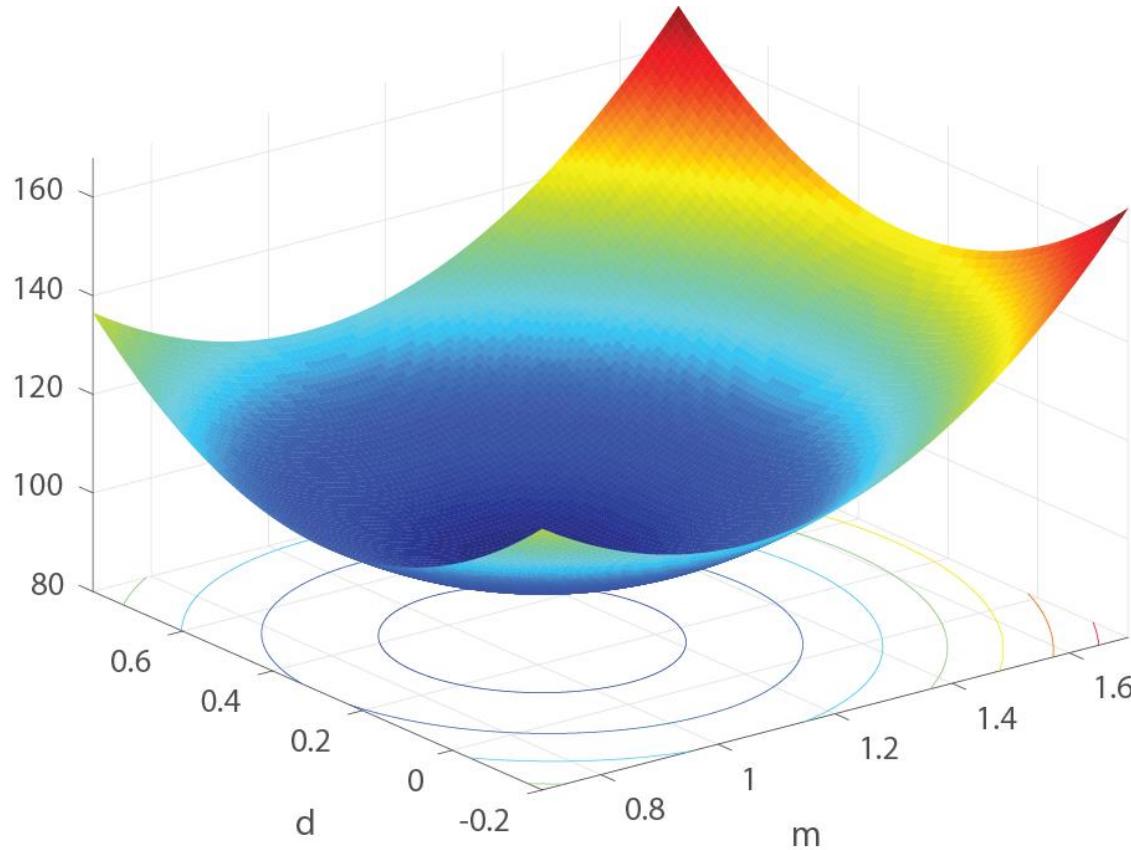
Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$\frac{\partial E}{\partial m} =$$

$$\frac{\partial E}{\partial d} =$$

Line Fitting



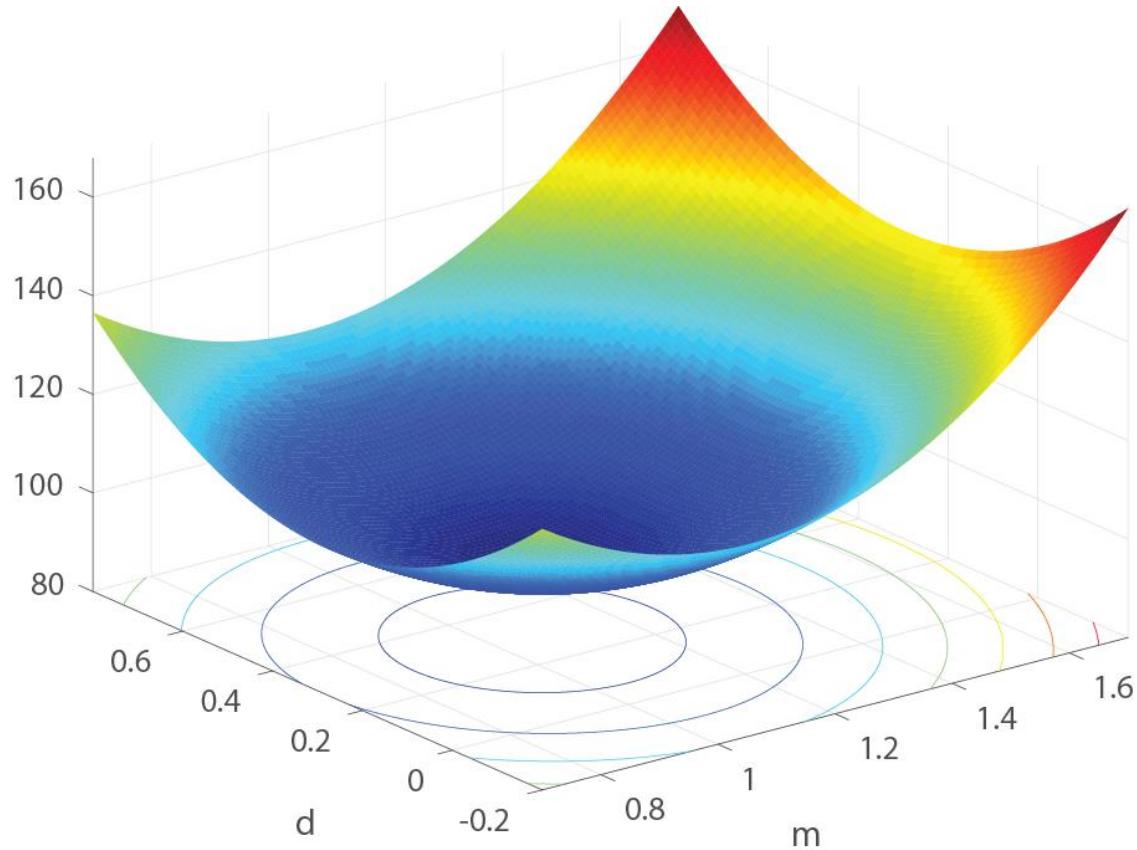
Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$\frac{\partial E}{\partial m} = -\sum_{i=1}^n 2u_i(v_i - (mu_i + d)) = 0$$

$$\frac{\partial E}{\partial d} = -\sum_{i=1}^n 2(v_i - (mu_i + d)) = 0$$

Line Fitting



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$\frac{\partial E}{\partial m} = -\sum_{i=1}^n 2u_i(v_i - (mu_i + d)) = 0$$

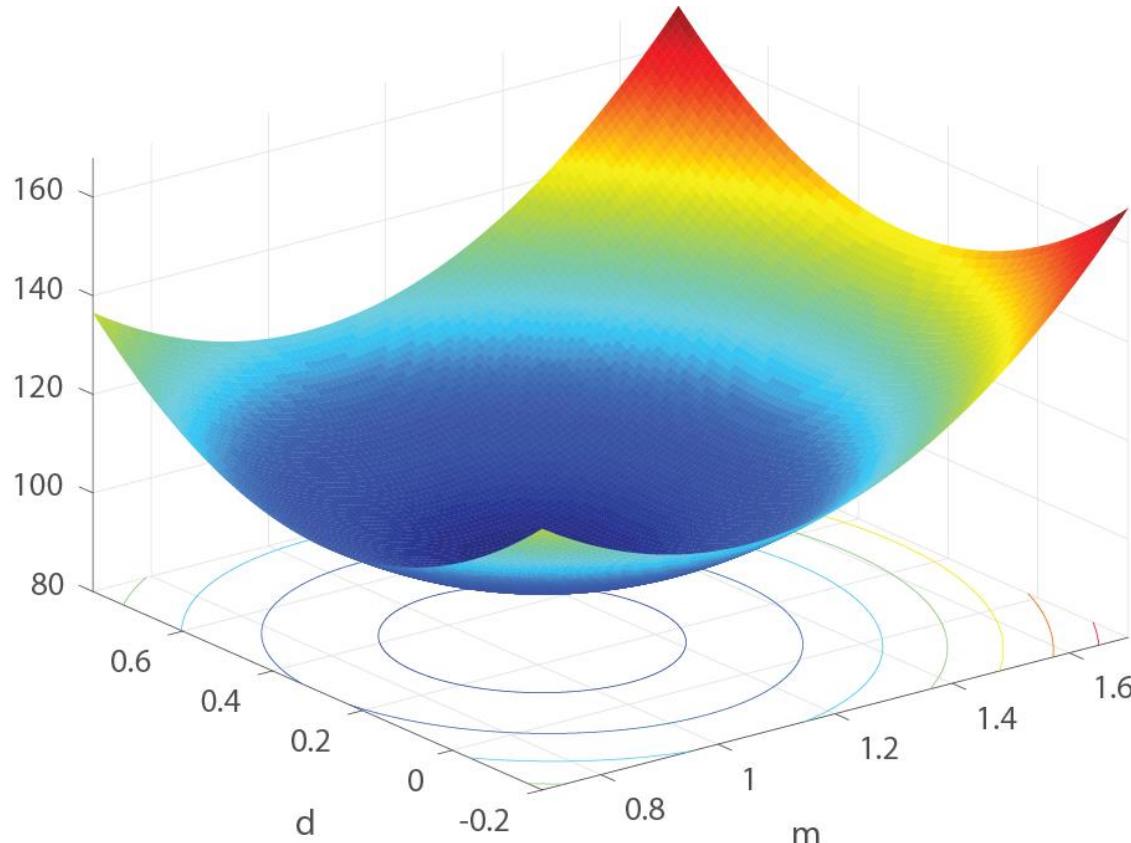
$$\frac{\partial E}{\partial d} = -\sum_{i=1}^n 2(v_i - (mu_i + d)) = 0$$



$$m \sum_{i=1}^n u_i^2 + d \sum_{i=1}^n u_i = \sum_{i=1}^n u_i v_i$$

$$m \sum_{i=1}^n u_i + n d = \sum_{i=1}^n v_i$$

Line Fitting



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$\frac{\partial E}{\partial m} = -\sum_{i=1}^n 2u_i(v_i - (mu_i + d)) = 0$$

$$\frac{\partial E}{\partial d} = -\sum_{i=1}^n 2(v_i - (mu_i + d)) = 0$$

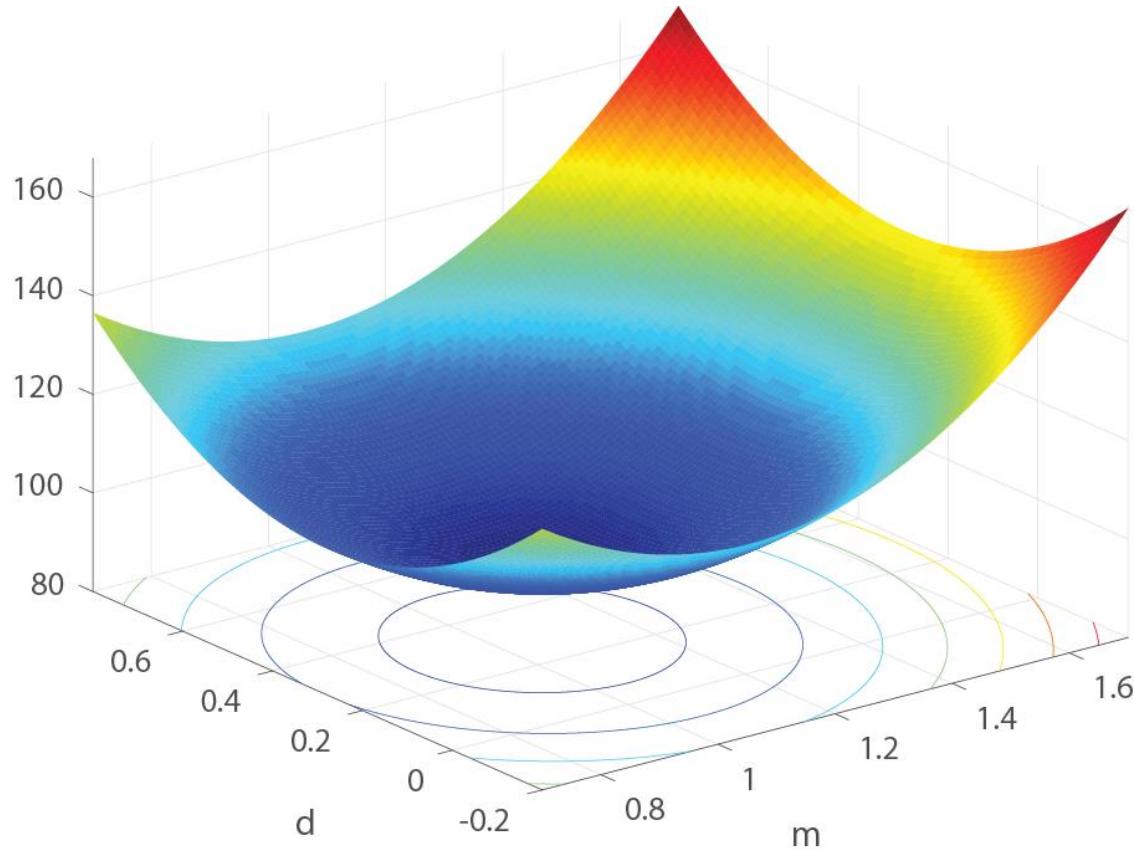
$$m \sum_{i=1}^n u_i^2 + d \sum_{i=1}^n u_i = \sum_{i=1}^n u_i v_i$$

$$m \sum_{i=1}^n u_i + n d = \sum_{i=1}^n v_i$$

$$\begin{bmatrix} \sum_{i=1}^n u_i^2 & \sum_{i=1}^n u_i \\ \sum_{i=1}^n u_i & n \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n u_i v_i \\ \sum_{i=1}^n v_i \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} m \\ d \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n u_i^2 & \sum_{i=1}^n u_i \\ \sum_{i=1}^n u_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n u_i v_i \\ \sum_{i=1}^n v_i \end{bmatrix}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

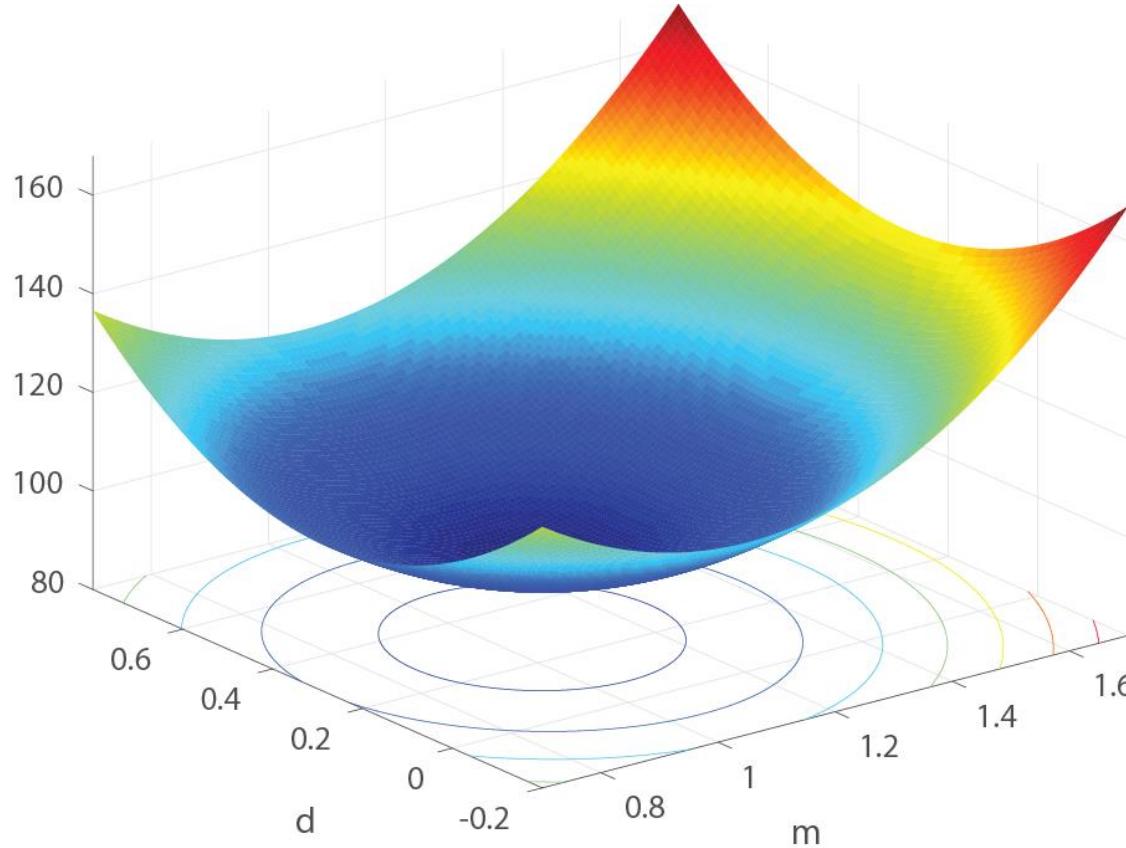
Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

$$v_n \approx mu_n + d$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

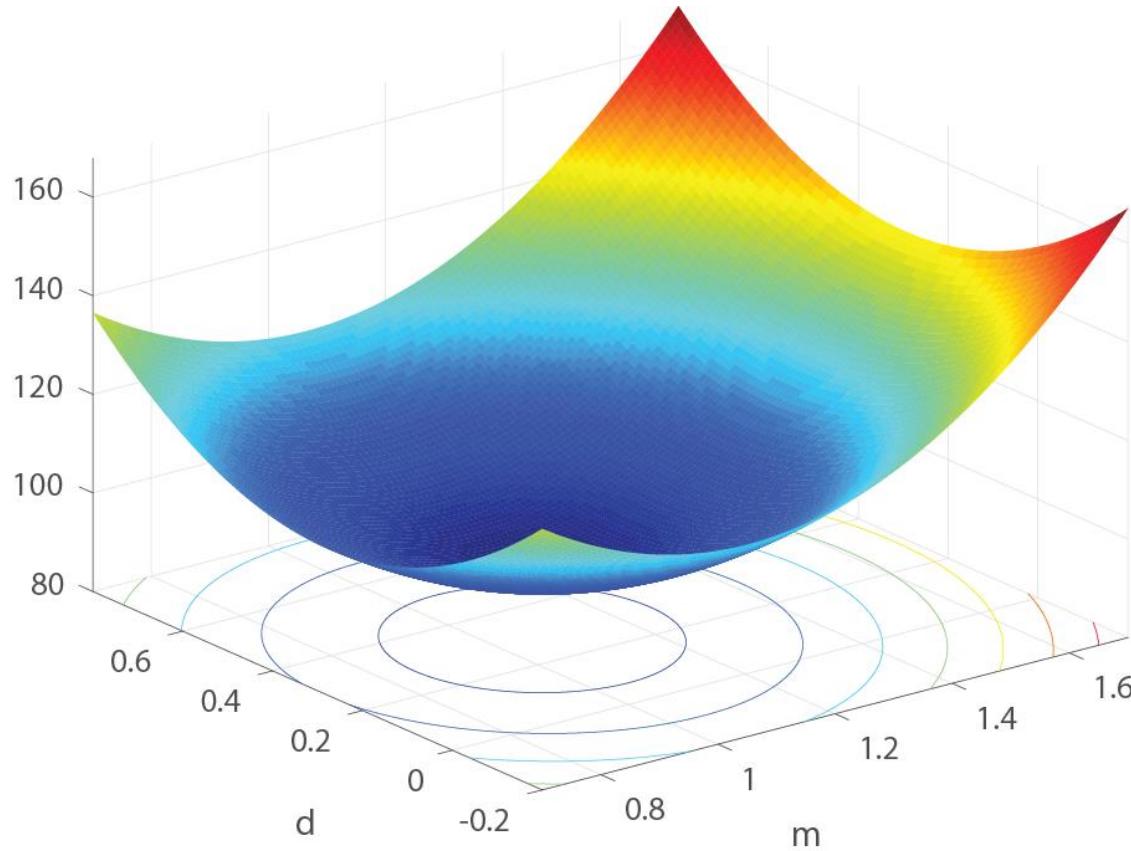
$$v_n \approx mu_n + d$$

$$\begin{bmatrix} u_1 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_1$$

$$\begin{bmatrix} u_2 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_2$$

$$\begin{bmatrix} u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_n$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

$$v_n \approx mu_n + d$$

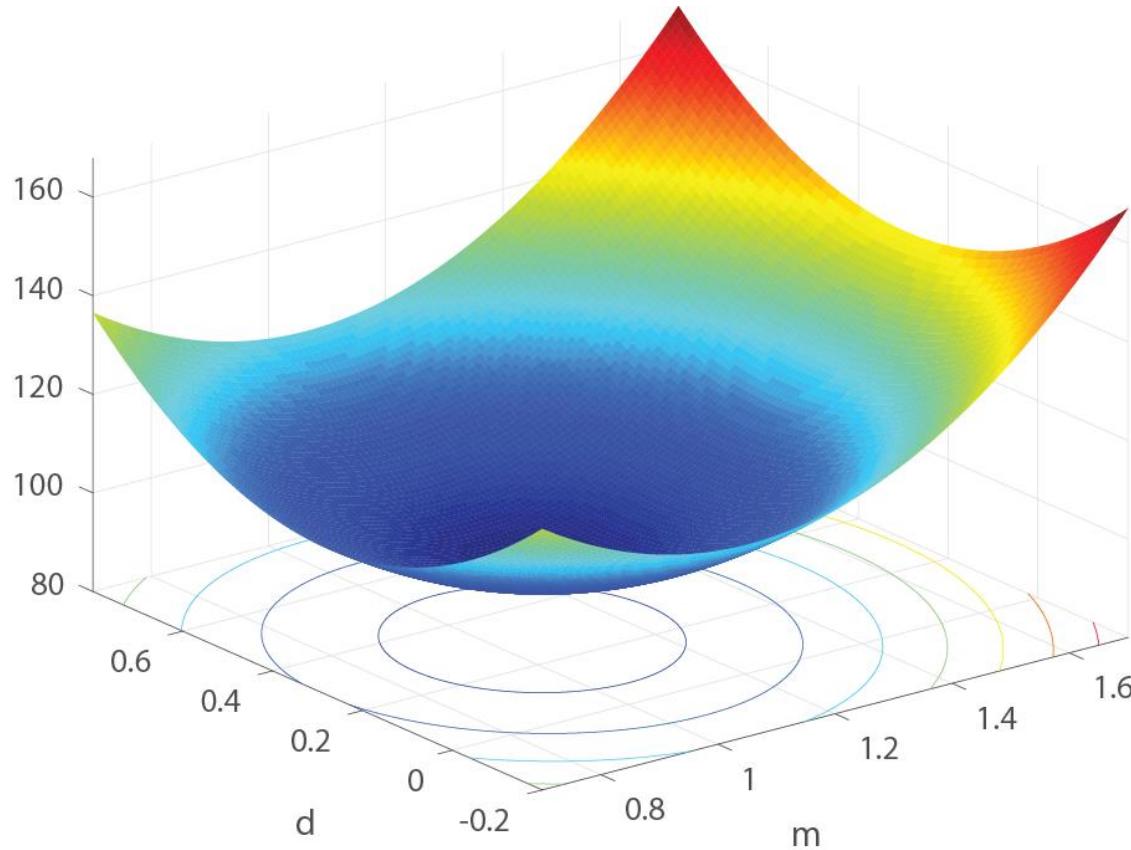
$$\begin{bmatrix} u_1 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_1$$

$$\begin{bmatrix} u_2 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_2$$

$$\begin{bmatrix} u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_n$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

$$v_n \approx mu_n + d$$

$$\begin{bmatrix} u_1 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_1$$

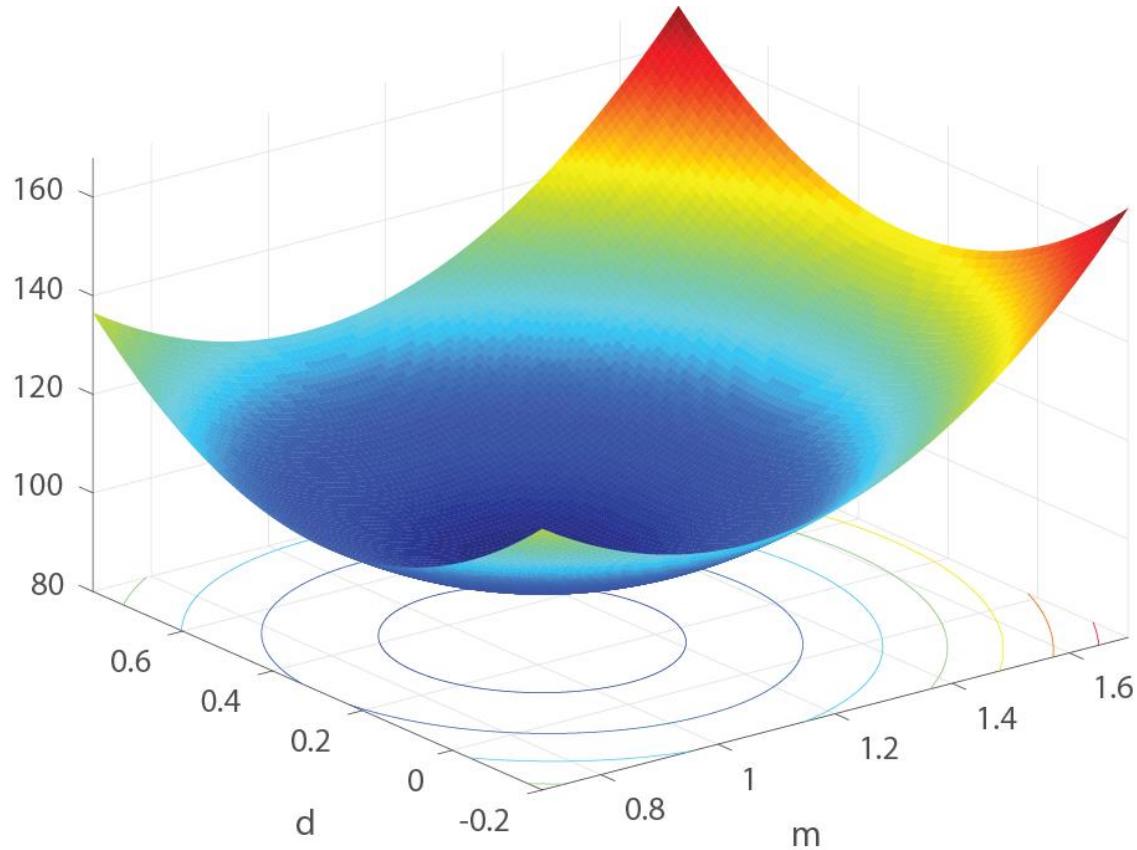
$$\begin{bmatrix} u_2 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_2$$

$$\begin{bmatrix} u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_n$$

$$\begin{bmatrix} u_1 & u_2 & \dots & u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

We can't invert **A**.

Line Fitting ($Ax=b$)

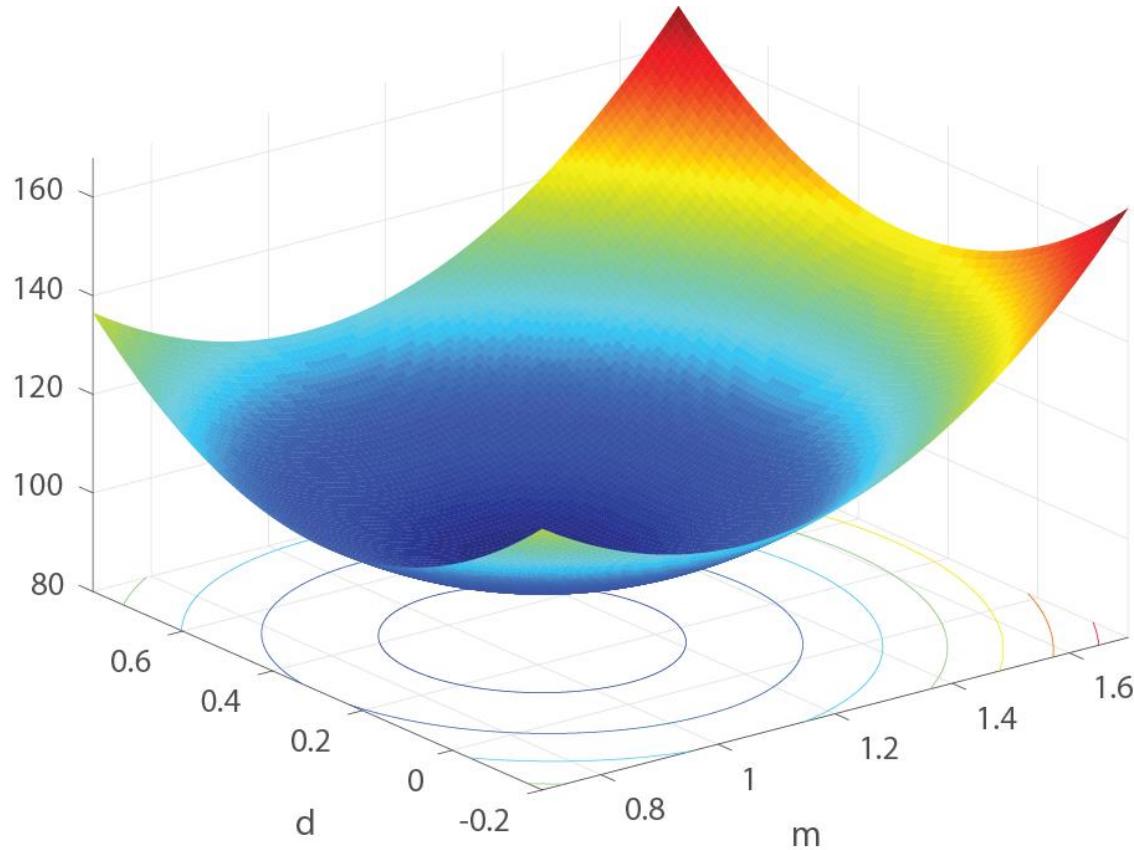


Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$E = (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



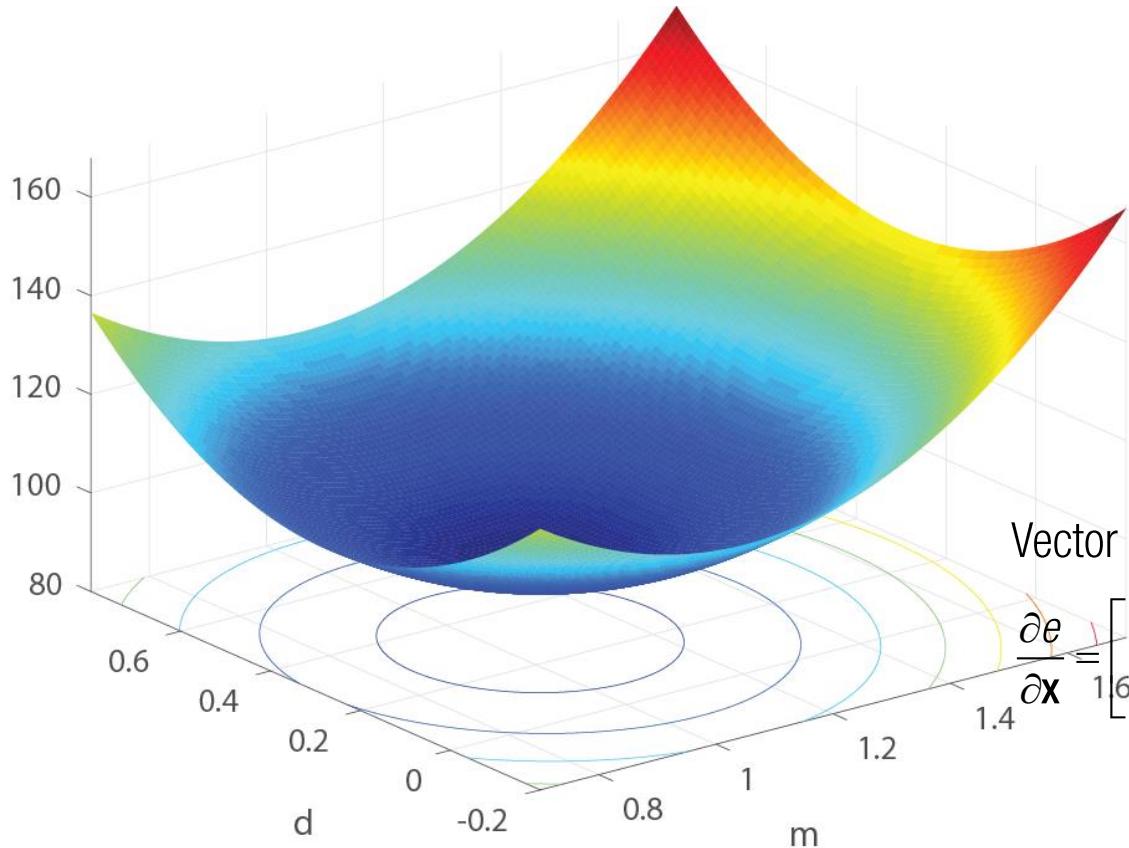
Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned}E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\&= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b}\end{aligned}$$

$$\frac{\partial E}{\partial \mathbf{x}} = ?$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \mathbf{b}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned}E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\&= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b}\end{aligned}$$

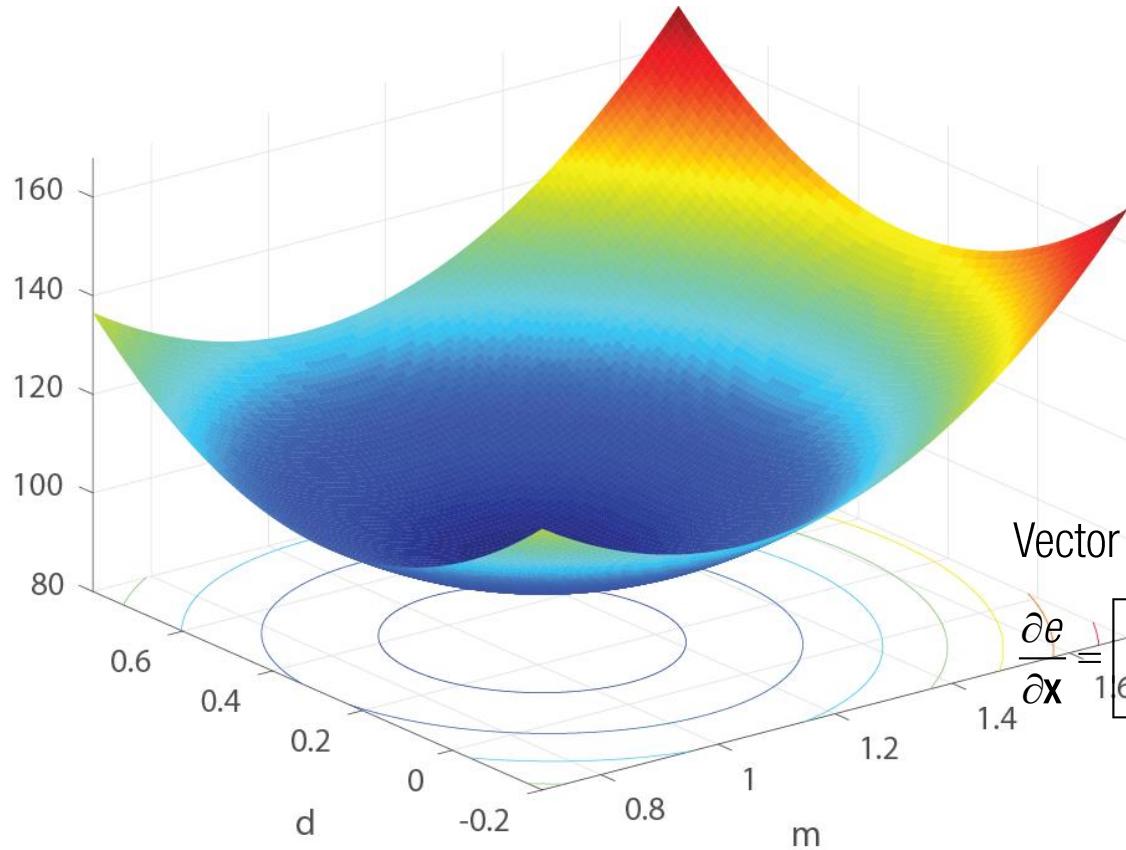
$$\frac{\partial E}{\partial \mathbf{x}} = ?$$

Vector derivative:

$$\frac{\partial e}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial e}{\partial x_1} & \dots & \frac{\partial e}{\partial x_n} \end{bmatrix}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned} E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\ &= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b} \end{aligned}$$

$$\frac{\partial E}{\partial \mathbf{x}} = ?$$

Vector derivative:

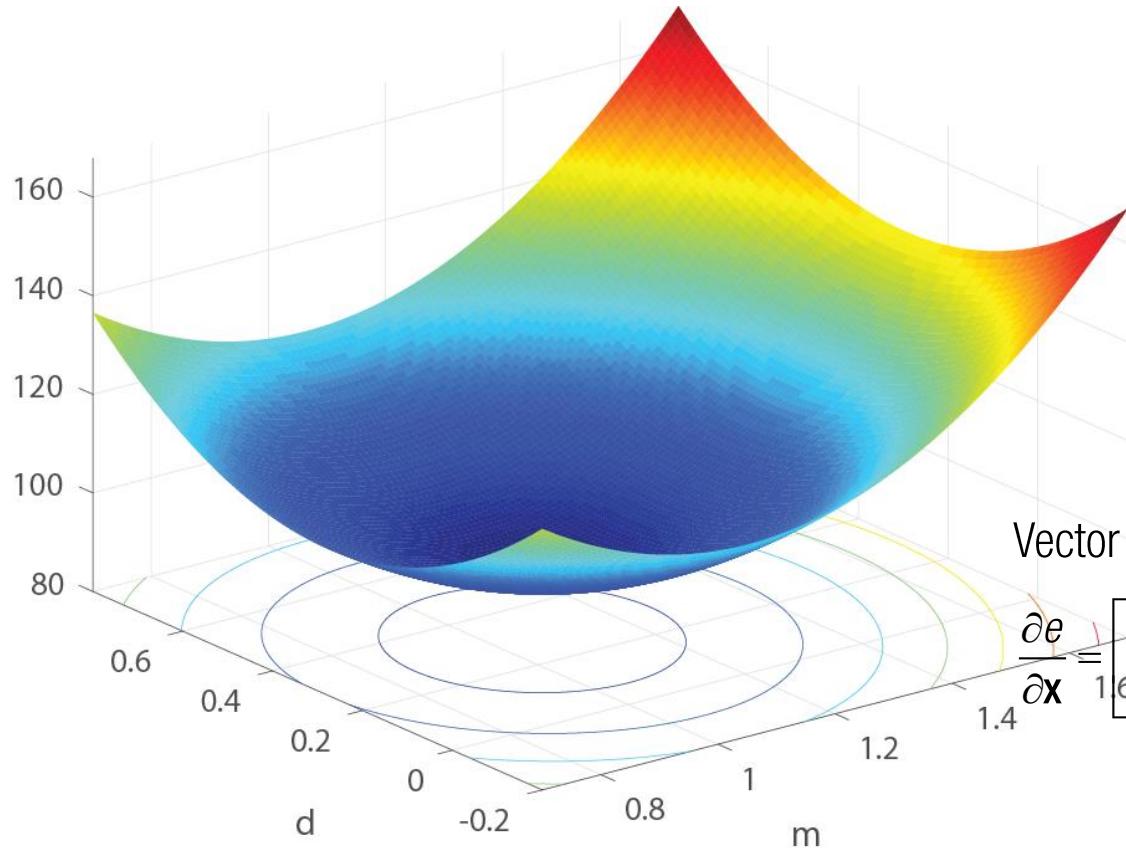
$$\frac{\partial e}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial e}{\partial x_1} & \dots & \frac{\partial e}{\partial x_n} \end{bmatrix}$$

Ex) $e = \mathbf{c}^\top \mathbf{x} = [c_1 \quad \dots \quad c_n] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

$$\frac{\partial \mathbf{c}^\top \mathbf{x}}{\partial \mathbf{x}} =$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$E = (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2$$

$$= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b}$$

$$\frac{\partial E}{\partial \mathbf{x}} = ?$$

Vector derivative:

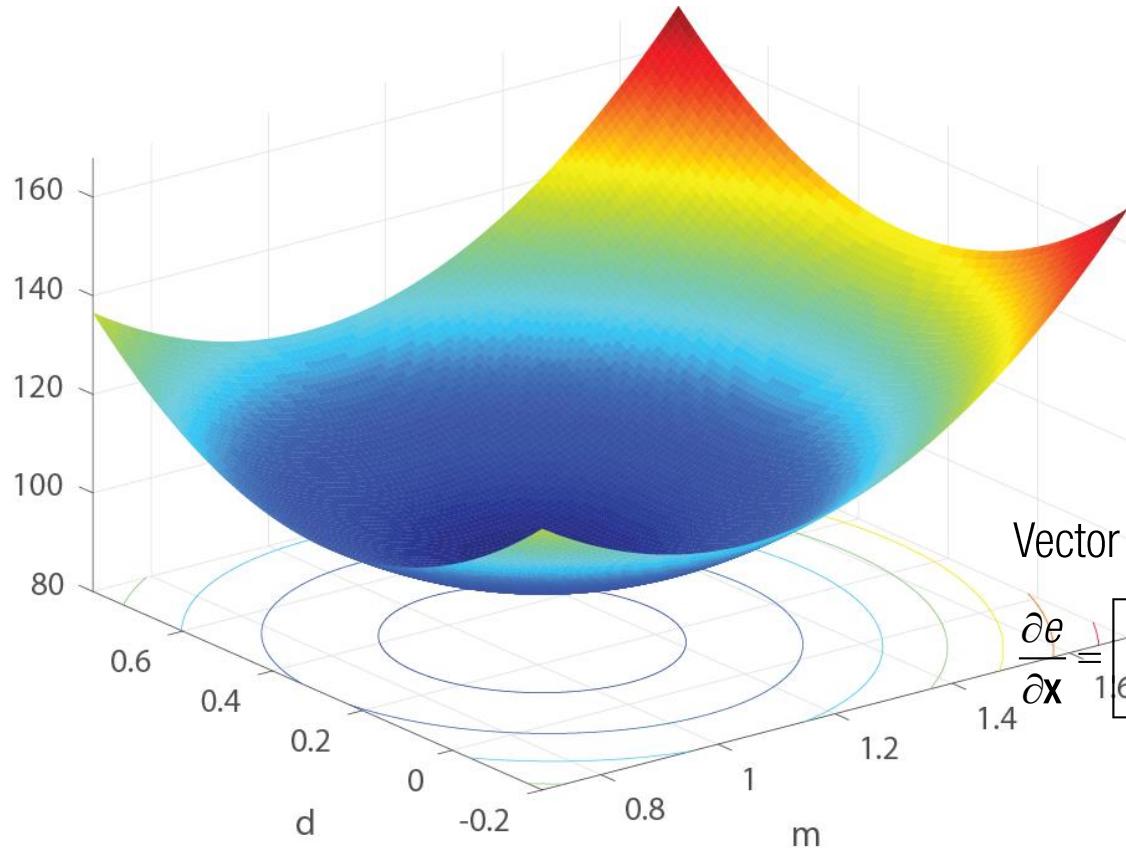
$$\frac{\partial e}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial e}{\partial x_1} & \dots & \frac{\partial e}{\partial x_n} \end{bmatrix}$$

Ex) $e = \mathbf{c}^\top \mathbf{x} = [c_1 \quad \dots \quad c_n] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

$$\frac{\partial \mathbf{c}^\top \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} (c_1 x_1 + \dots + c_n x_n)$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned} E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\ &= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b} \end{aligned}$$

$$\frac{\partial E}{\partial \mathbf{x}} = ?$$

Vector derivative:

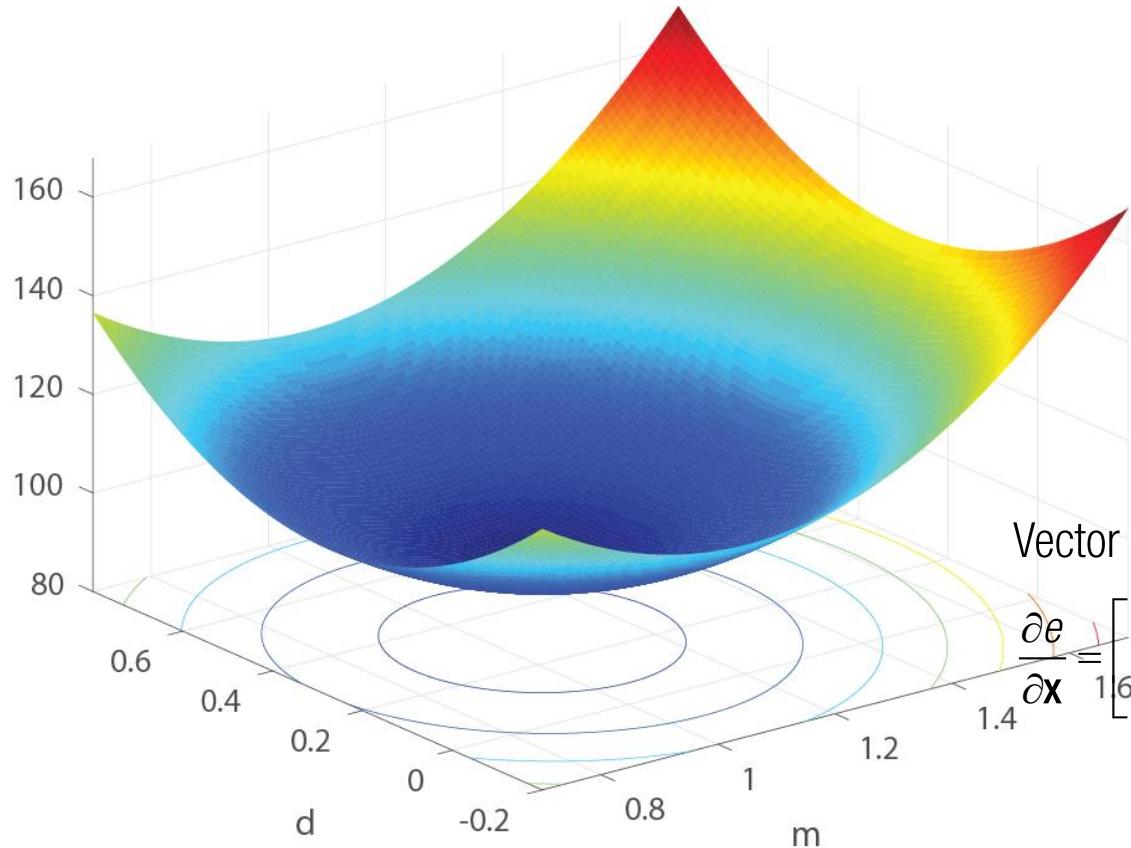
$$\frac{\partial e}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial e}{\partial x_1} & \dots & \frac{\partial e}{\partial x_n} \end{bmatrix}$$

Ex) $e = \mathbf{c}^\top \mathbf{x} = [c_1 \quad \dots \quad c_n] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

$$\begin{aligned} \frac{\partial \mathbf{c}^\top \mathbf{x}}{\partial \mathbf{x}} &= \frac{\partial}{\partial \mathbf{x}} (c_1 x_1 + \dots + c_n x_n) \\ &= [c_1 \quad \dots \quad c_n] \end{aligned}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned} E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\ &= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b} \end{aligned}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \mathbf{b}$$

$$\frac{\partial E}{\partial \mathbf{x}} = 2\mathbf{A}^\top \mathbf{Ax} - 2\mathbf{A}^\top \mathbf{b} = \mathbf{0}$$

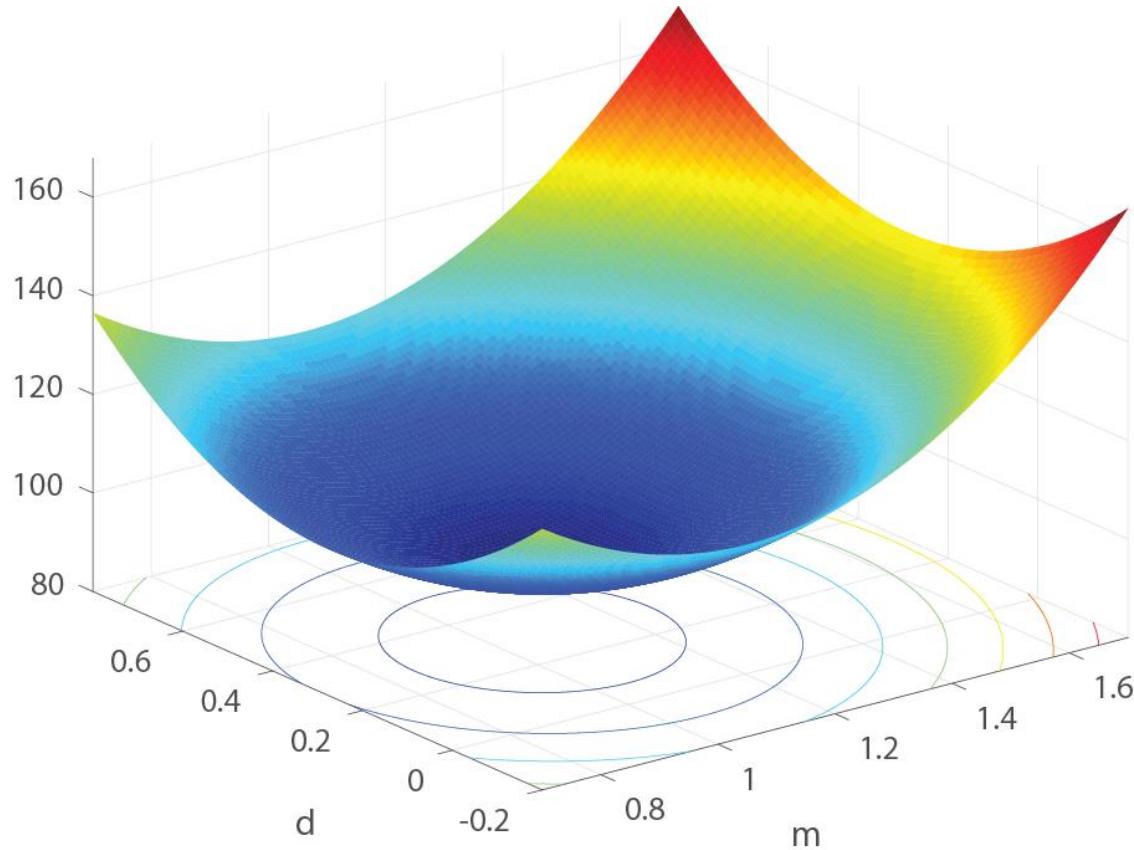
Vector derivative:

$$\frac{\partial e}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial e}{\partial x_1} & \dots & \frac{\partial e}{\partial x_n} \end{bmatrix}$$

Ex) $e = \mathbf{c}^\top \mathbf{x} = \begin{bmatrix} c_1 & \dots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

$$\begin{aligned} \frac{\partial \mathbf{c}^\top \mathbf{x}}{\partial \mathbf{x}} &= \frac{\partial}{\partial \mathbf{x}} (c_1 x_1 + \dots + c_n x_n) \\ &= [c_1 \quad \dots \quad c_n] \end{aligned}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

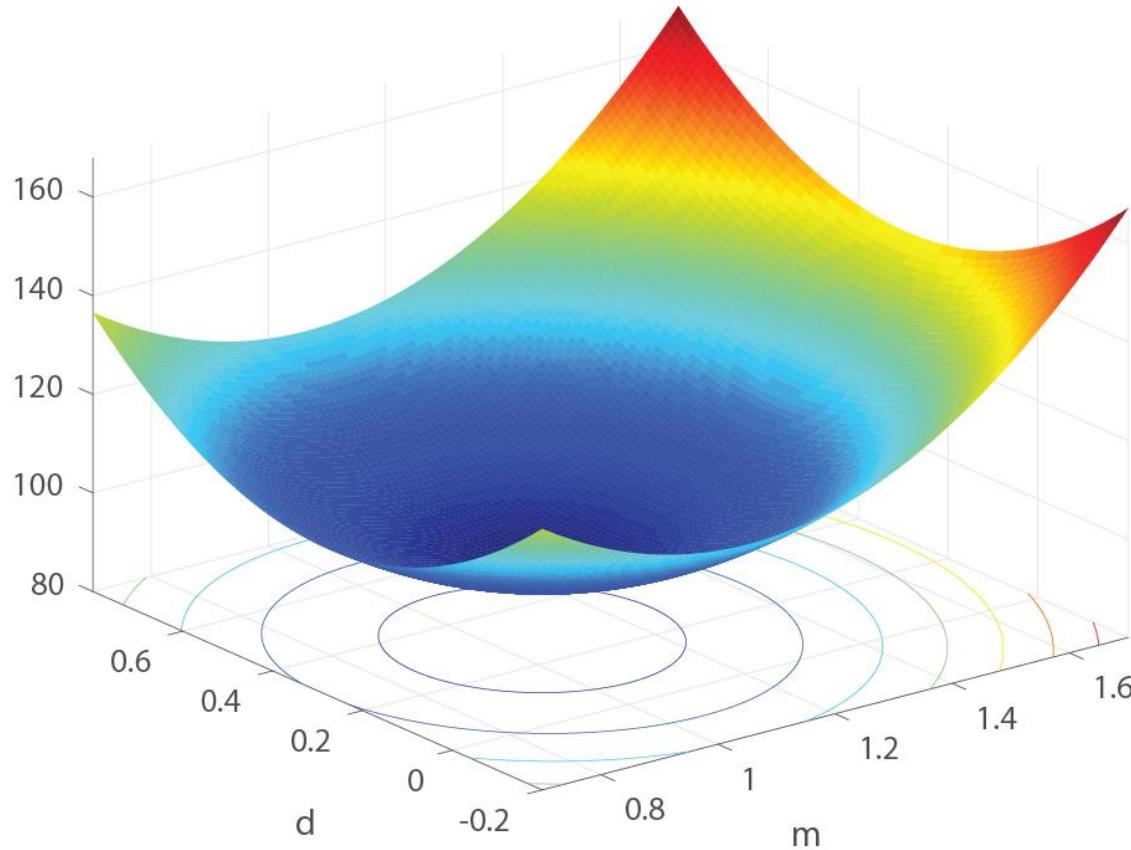
$$\begin{aligned}E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\&= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b}\end{aligned}$$

$$\frac{\partial E}{\partial \mathbf{x}} = 2\mathbf{A}^\top \mathbf{Ax} - 2\mathbf{A}^\top \mathbf{b} = \mathbf{0}$$

$$\rightarrow \mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{b}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} \mathbf{b} \end{bmatrix}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned} E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\ &= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b} \end{aligned}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{matrix} \mathbf{A} \\ \mathbf{x} \\ \vdots \\ \mathbf{b} \end{matrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

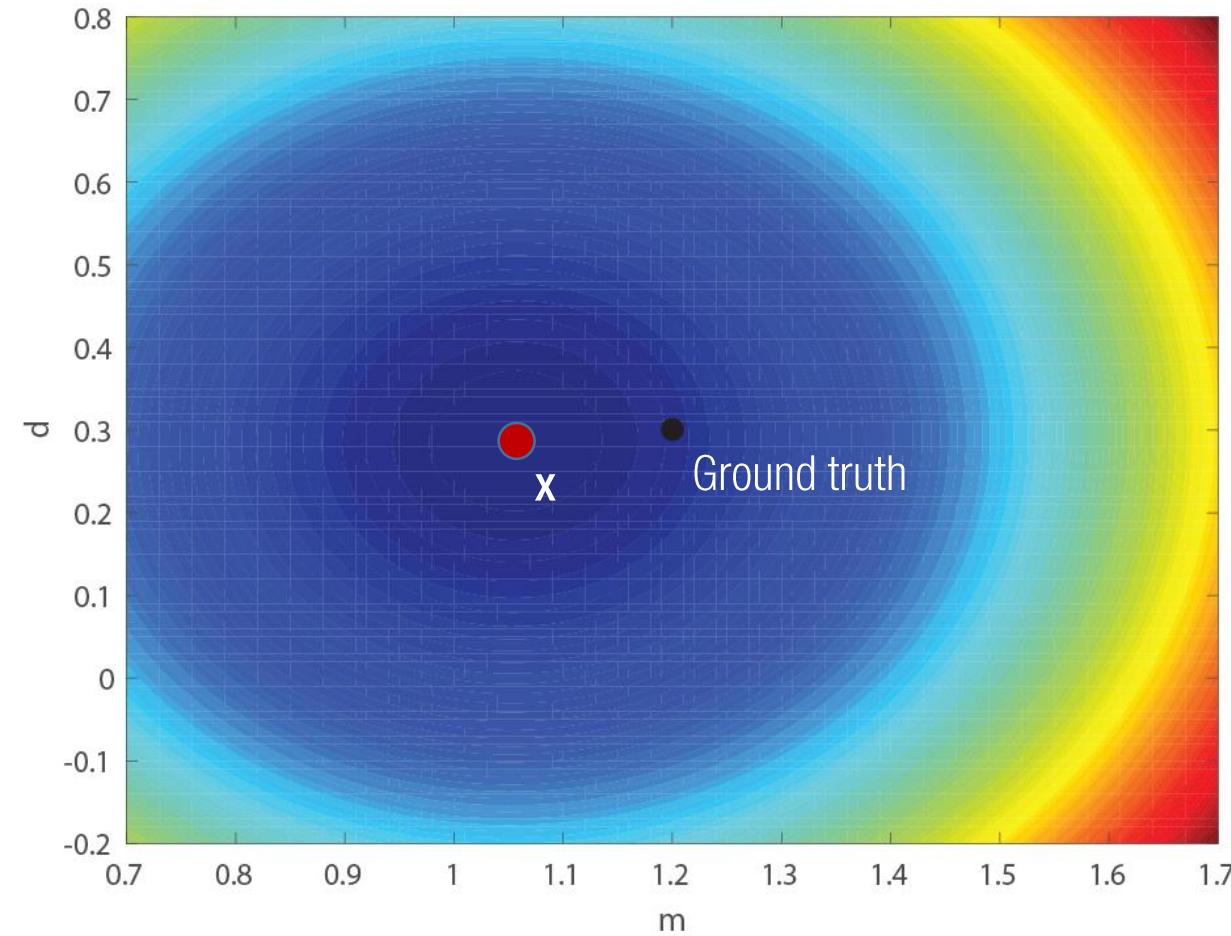
$$\frac{\partial E}{\partial \mathbf{x}} = 2\mathbf{A}^\top \mathbf{Ax} - 2\mathbf{A}^\top \mathbf{b} = \mathbf{0}$$

$$\rightarrow \mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{b}$$

$$\begin{array}{c} \mathbf{A}^\top \\ \mathbf{A} \\ \mathbf{x} \end{array} = \begin{array}{c} \mathbf{A}^\top \\ \mathbf{A} \\ \mathbf{b} \end{array}$$

Normal equation

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned}E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\&= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b}\end{aligned}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} \mathbf{b} \end{bmatrix}$$

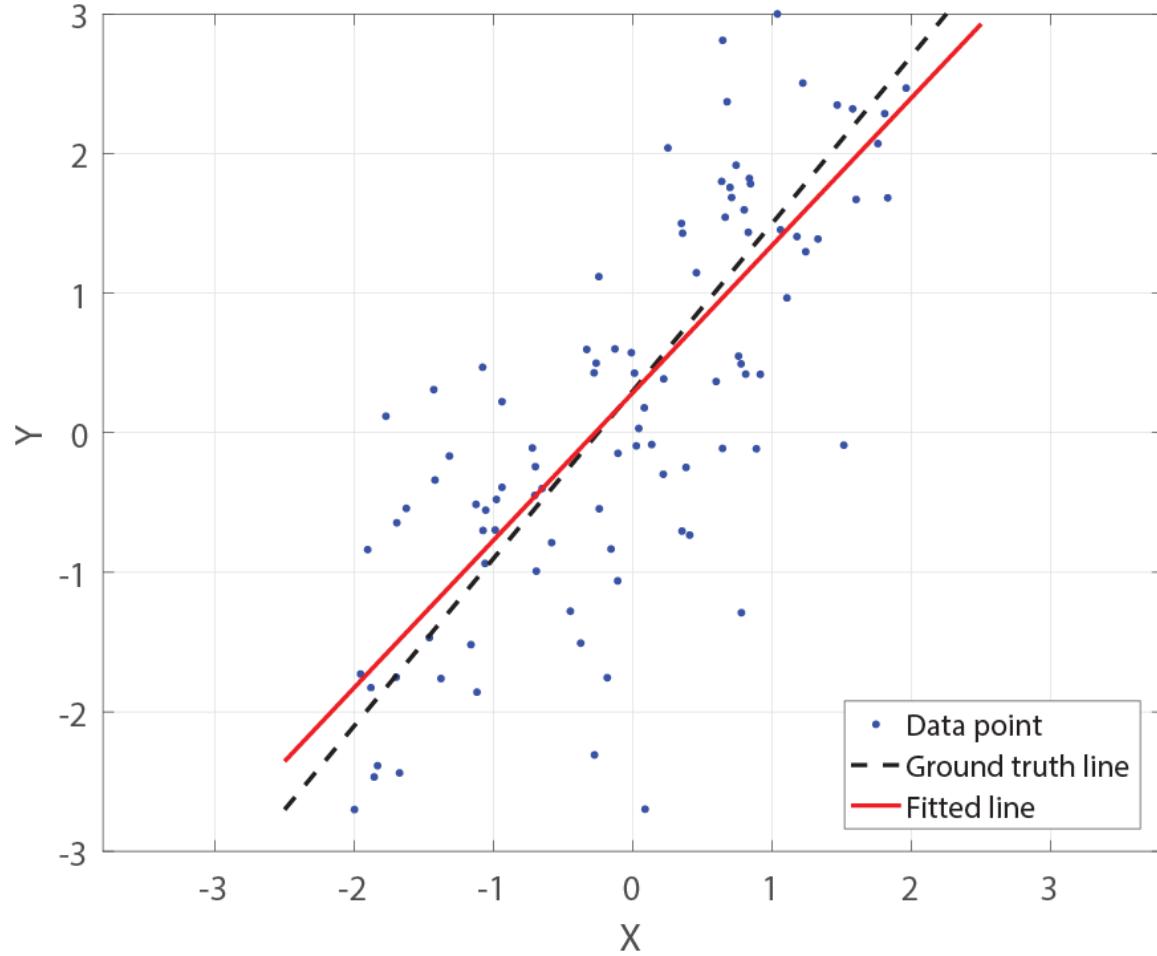
$$\frac{\partial E}{\partial \mathbf{x}} = 2\mathbf{A}^\top \mathbf{Ax} - 2\mathbf{A}^\top \mathbf{b} = \mathbf{0}$$

$$\rightarrow \mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{b}$$

$$\begin{array}{c} \mathbf{A}^\top \\ \mathbf{A} \\ \mathbf{x} \end{array} = \begin{array}{c} \mathbf{A}^\top \\ \mathbf{b} \end{array}$$

$$\mathbf{x} = \left[\begin{array}{cc} \mathbf{A}^\top & \mathbf{A} \end{array} \right]^{-1} \begin{array}{c} \mathbf{A}^\top \\ \mathbf{b} \end{array}$$

Line Fitting ($Ax=b$)



Total error: $E = \sum_{i=1}^n (v_i - (mu_i + d))^2$

$$\begin{aligned} E &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2 \\ &= \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - 2\mathbf{x}^\top \mathbf{A}^\top \mathbf{b} + \mathbf{b}^\top \mathbf{b} \end{aligned}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \approx \begin{bmatrix} m \\ d \end{bmatrix} \mathbf{b}$$

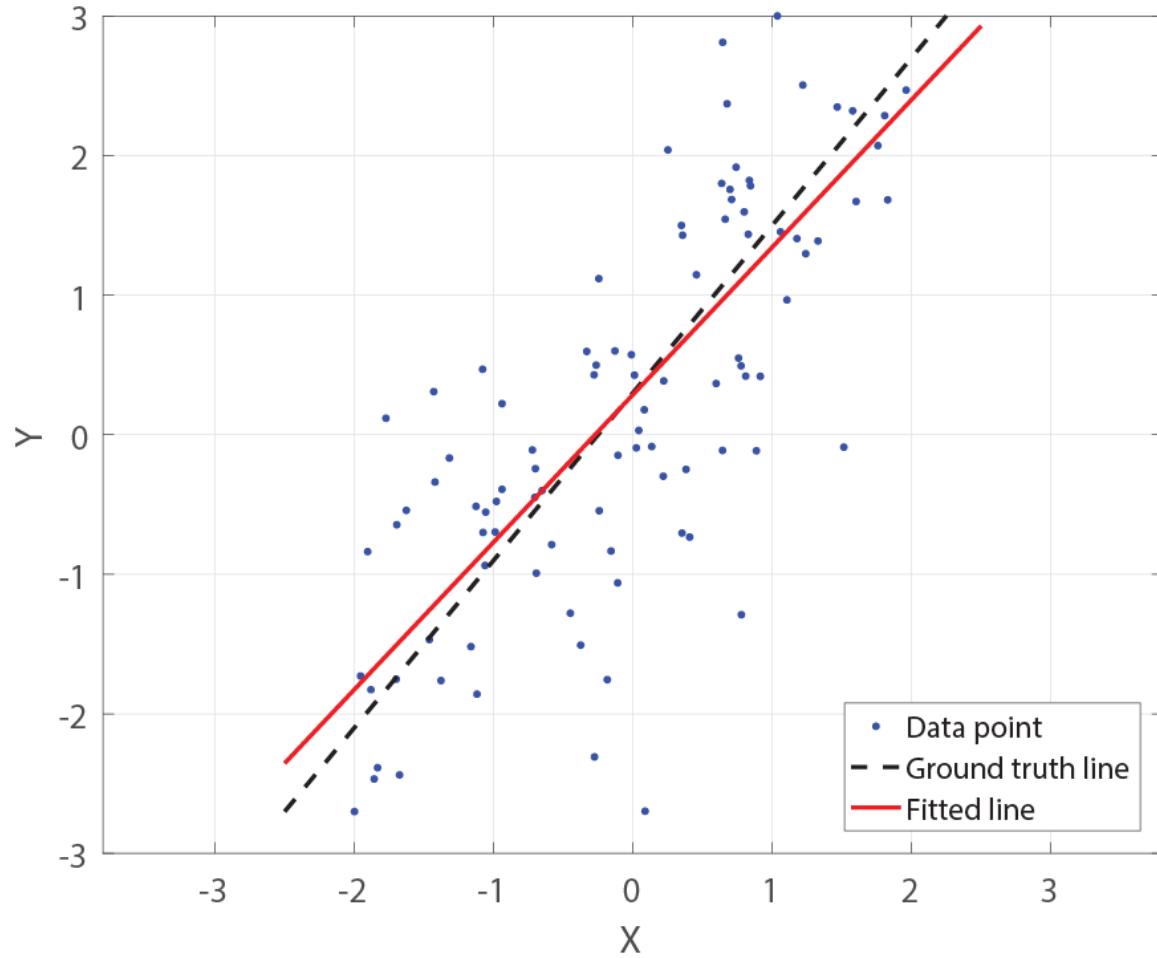
$$\frac{\partial E}{\partial \mathbf{x}} = 2\mathbf{A}^\top \mathbf{Ax} - 2\mathbf{A}^\top \mathbf{b} = \mathbf{0}$$

$$\rightarrow \mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{b}$$

$$\mathbf{A}^\top \mathbf{A} \mathbf{x} = \mathbf{A}^\top \mathbf{b}$$

$$\mathbf{x} = [\mathbf{A}^\top \quad \mathbf{A}]^{-1} \mathbf{A}^\top \mathbf{b}$$

Line Fitting ($Ax=b$)



LineFitting.m

$m = 1.2;$
 $d = 0.3;$

$x = 4 * (\text{rand}(100,1) - 0.5);$
 $y = m * x + d + \text{randn}(\text{size}(x));$

$A = [x \ ones(\text{size}(x))];$
 $b = y;$

$u = A \backslash b;$

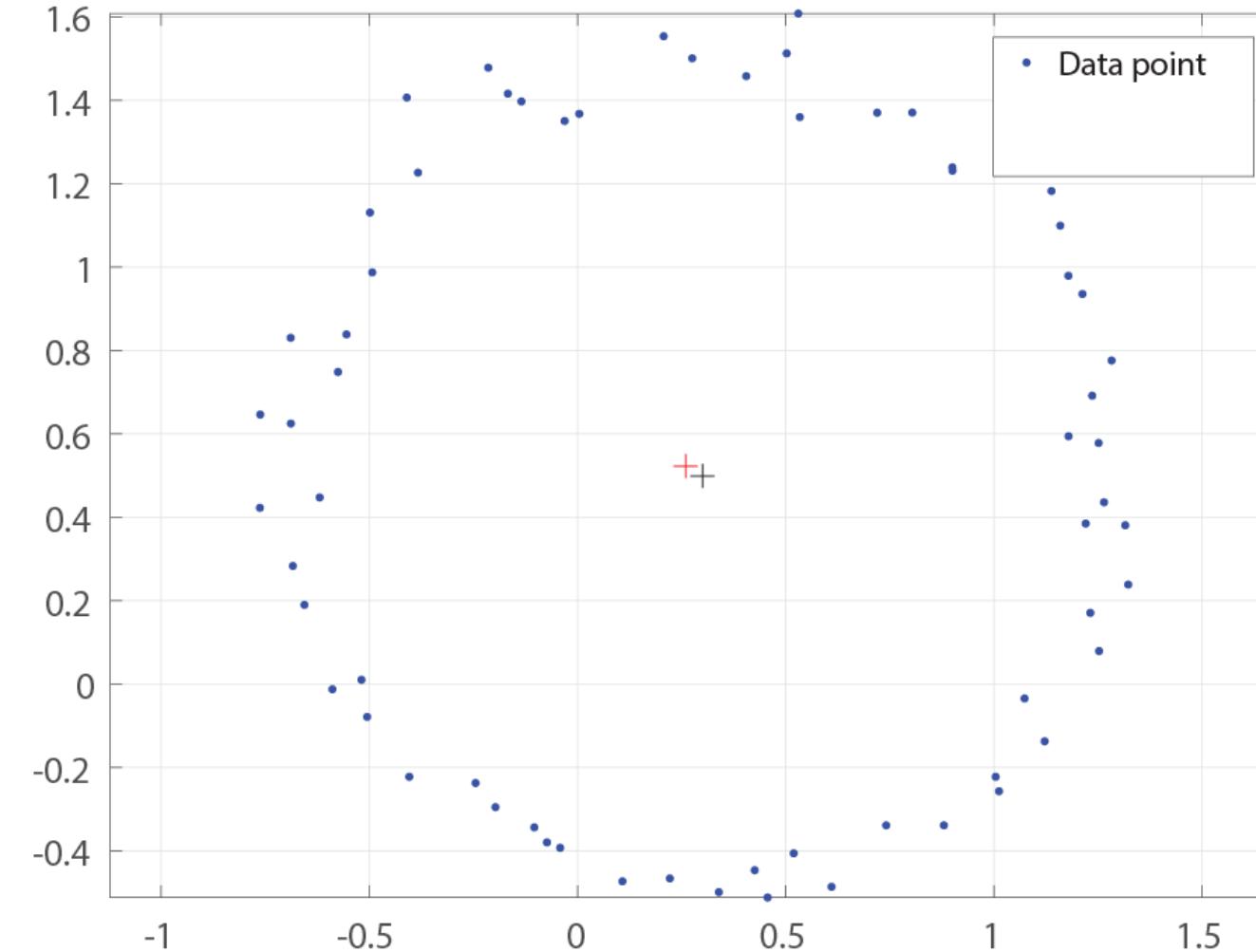
$$\leftarrow x = (A^T A)^{-1} A^T b$$

Ground truth

Random data point w/
Gaussian noise

\ backslash: solve linear least squares

Circle Fitting ($Ax=b$)



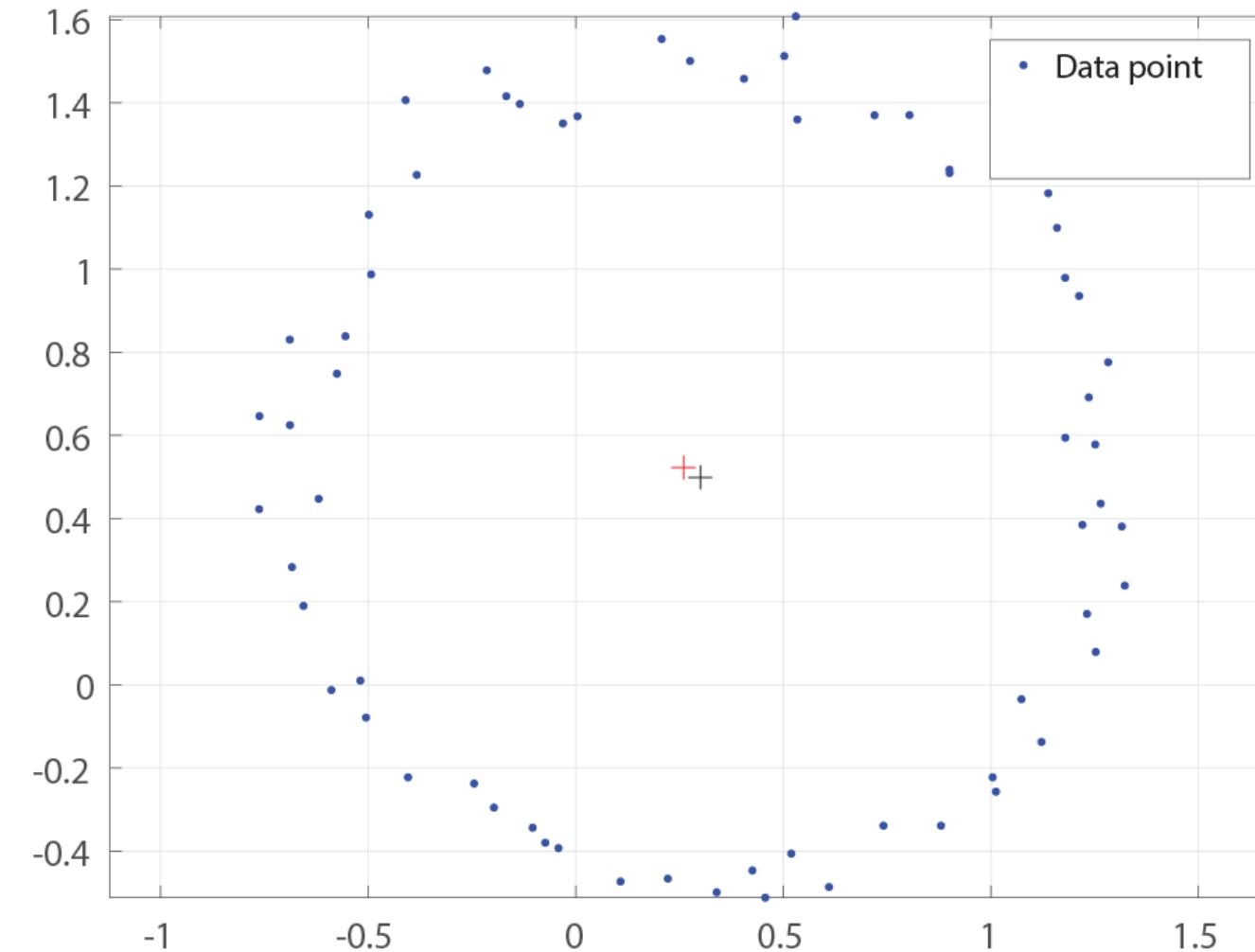
$$(x_1 - C_x)^2 + (y_1 - C_y)^2 = r^2$$

⋮

$$(x_n - C_x)^2 + (y_n - C_y)^2 = r^2$$

Unknowns: C_x, C_y, r

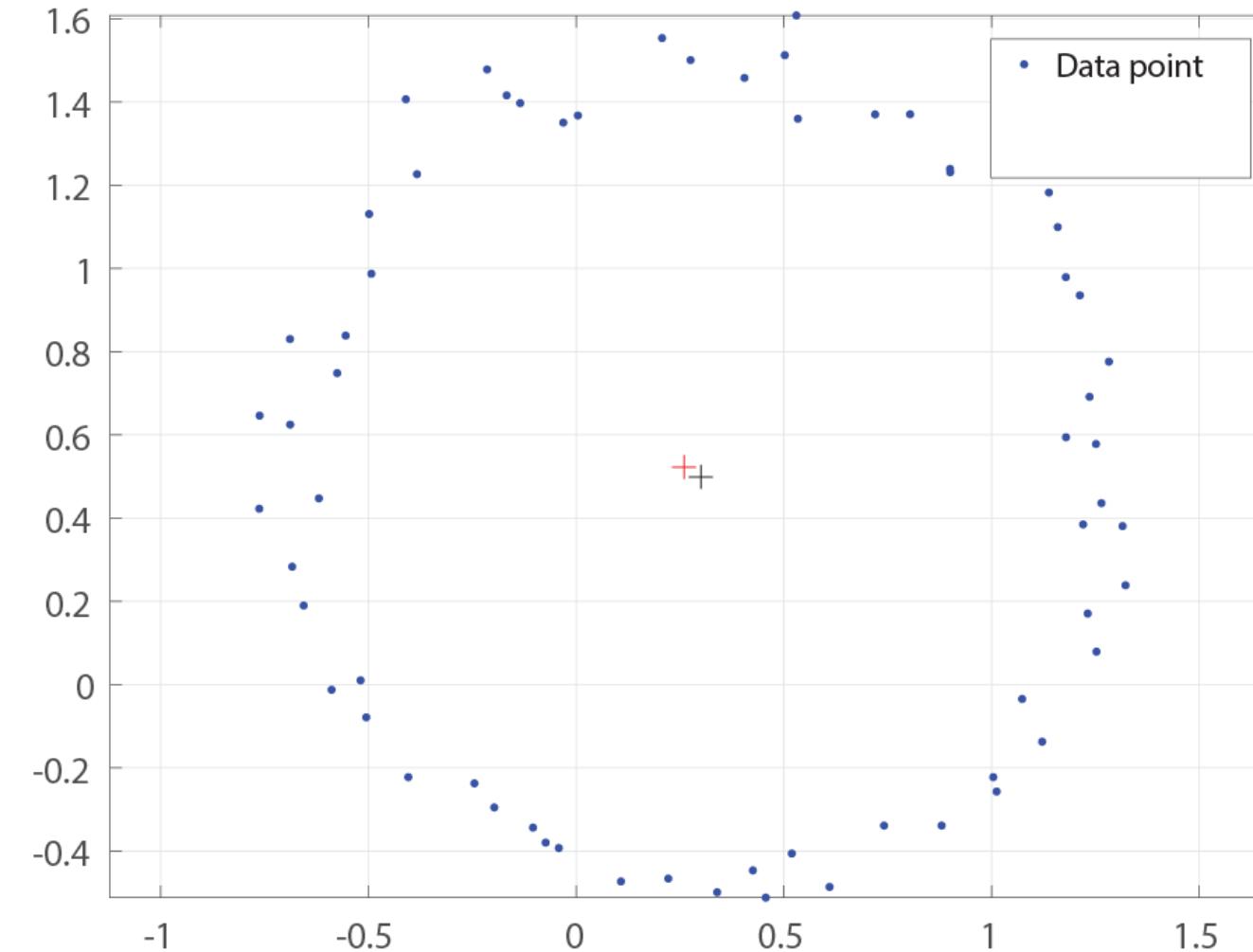
Circle Fitting ($Ax=b$)



$$x_1^2 - 2x_1 C_x + C_x^2 + y_1^2 - 2y_1 C_y + C_y^2 = r^2$$

$$x_n^2 - 2x_n C_x + C_x^2 + y_n^2 - 2y_n C_y + C_y^2 = r^2$$

Circle Fitting ($Ax=b$)

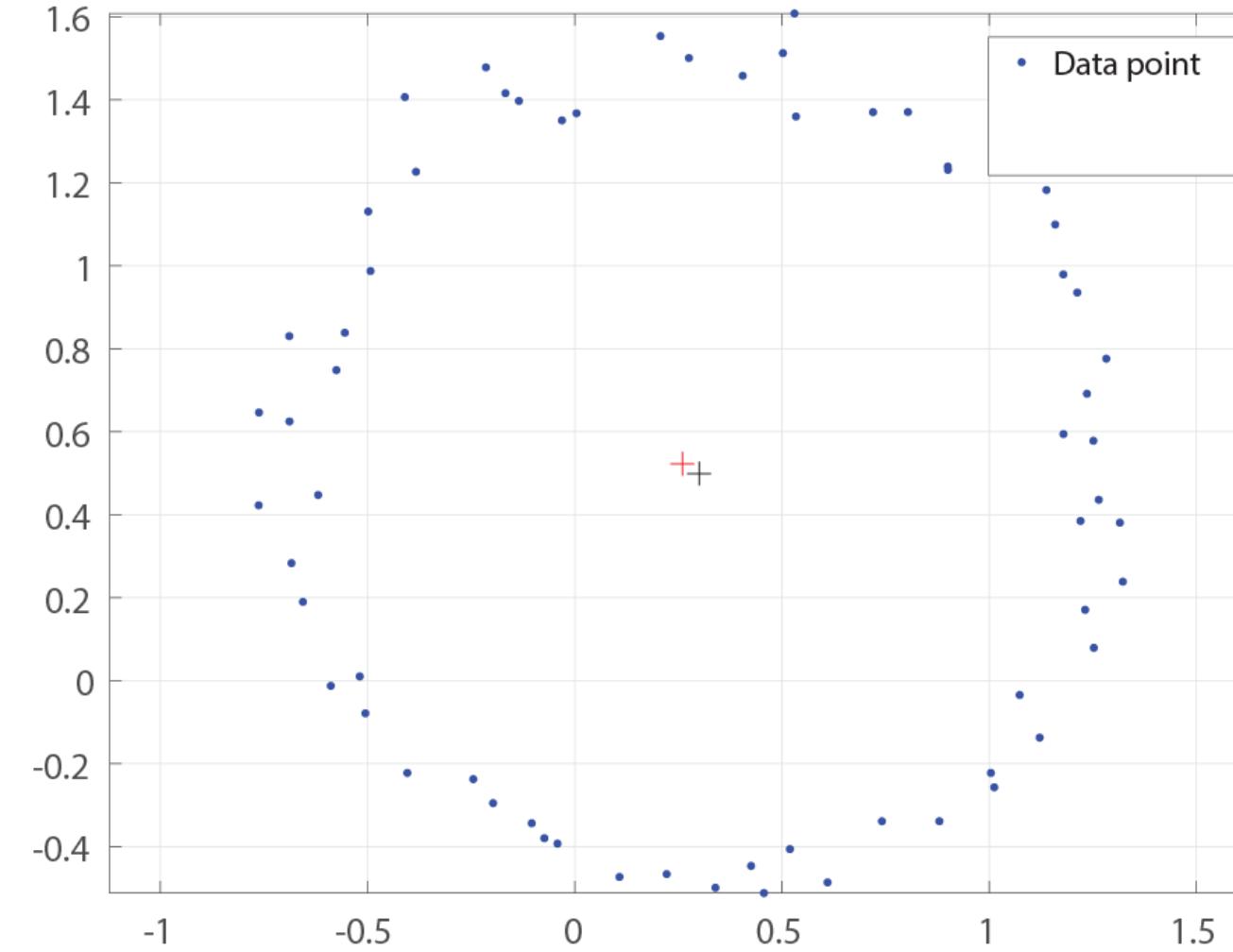


$$x_1^2 - 2x_1 C_x + C_x^2 + y_1^2 - 2y_1 C_y + C_y^2 = r^2$$

$$\vdots$$
$$x_n^2 - 2x_n C_x + C_x^2 + y_n^2 - 2y_n C_y + C_y^2 = r^2$$

$$x_i^2 - x_1^2 - 2C_x(x_i - x_1) + y_i^2 - y_1^2 - 2(y_i - y_1)C_y = 0$$

Circle Fitting ($Ax=b$)



$$x_1^2 - 2x_1 C_x + C_x^2 + y_1^2 - 2y_1 C_y + C_y^2 = r^2$$

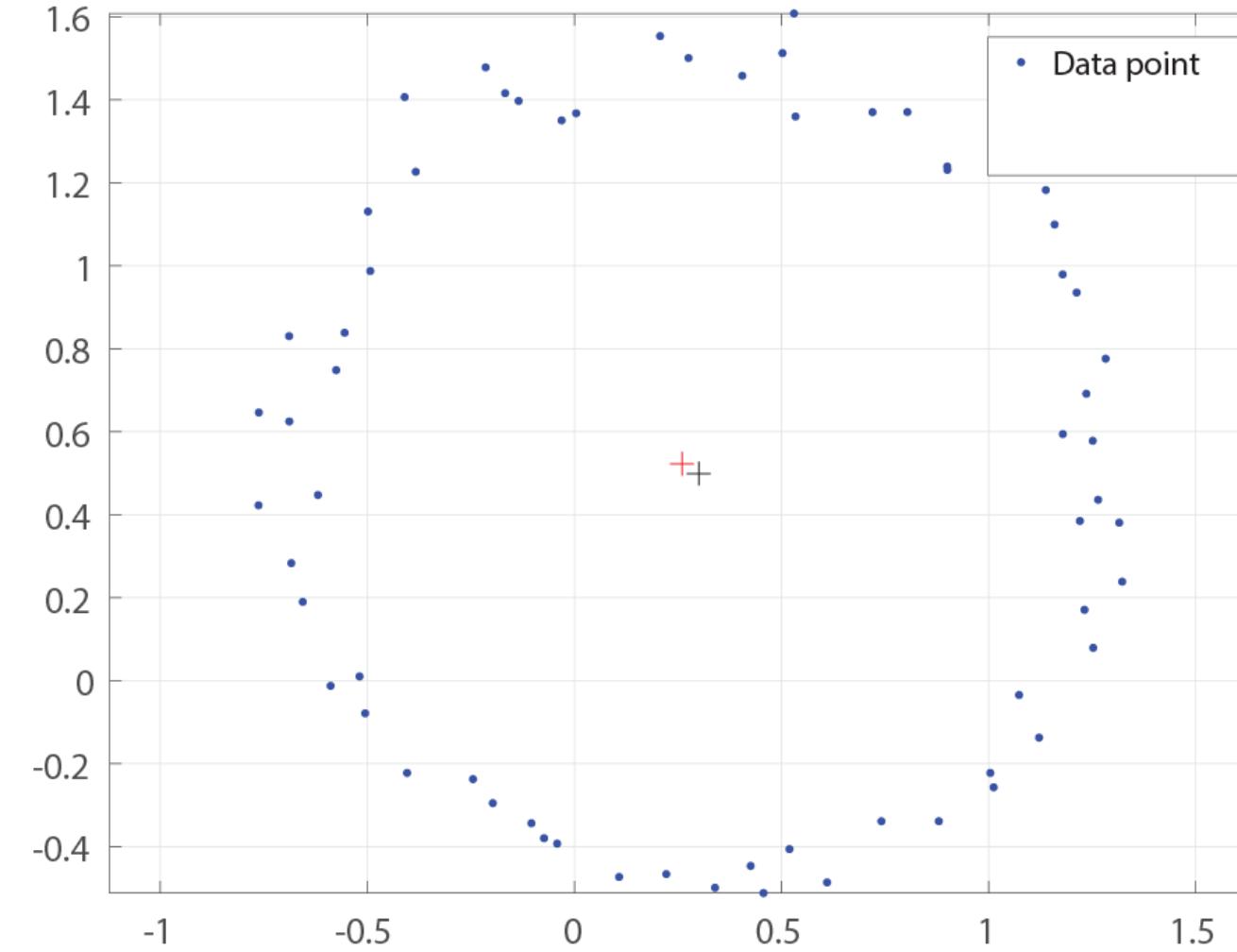
$$\vdots$$

$$x_n^2 - 2x_n C_x + C_x^2 + y_n^2 - 2y_n C_y + C_y^2 = r^2$$

$$x_i^2 - x_1^2 - 2C_x(x_i - x_1) + y_i^2 - y_1^2 - 2(y_i - y_1)C_y = 0$$

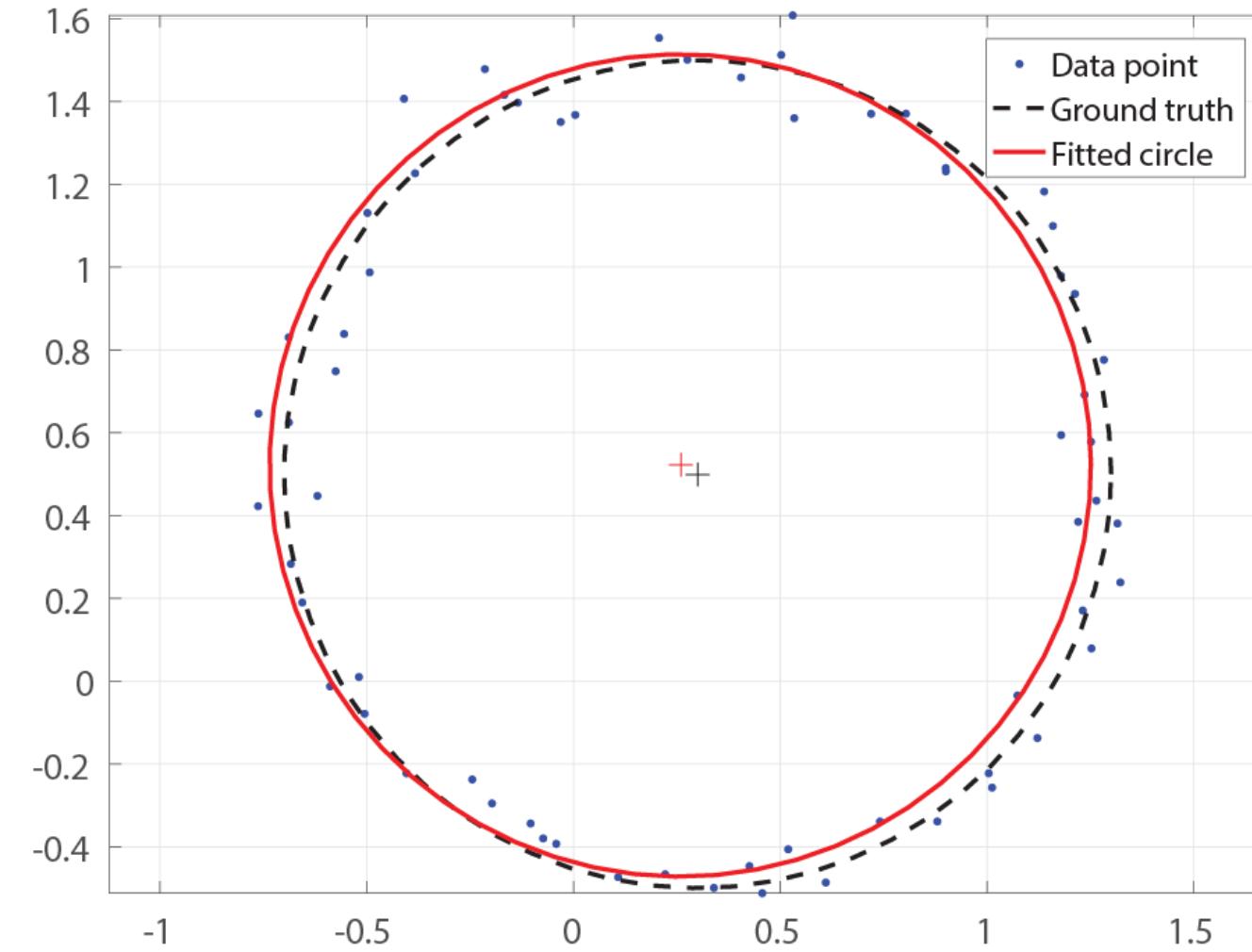
$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ \vdots & \vdots \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix} \begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \vdots \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}$$

Circle Fitting ($Ax=b$)



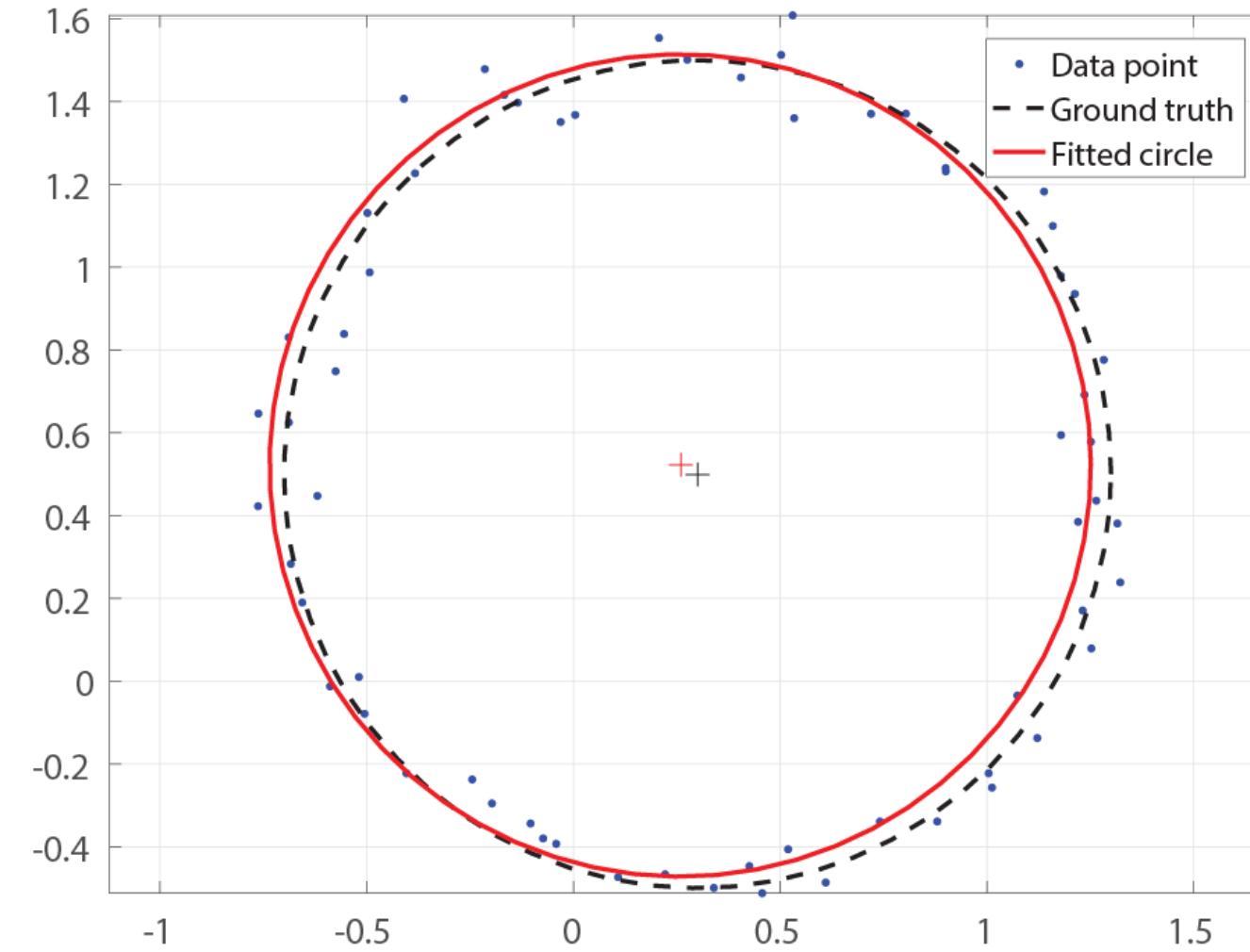
$$x_1^2 - 2x_1 C_x + C_x^2 + y_1^2 - 2y_1 C_y + C_y^2 = r^2$$
$$\vdots \quad \vdots$$
$$x_n^2 - 2x_n C_x + C_x^2 + y_n^2 - 2y_n C_y + C_y^2 = r^2$$
$$\downarrow$$
$$x_i^2 - x_1^2 - 2C_x(x_i - x_1) + y_i^2 - y_1^2 - 2(y_i - y_1)C_y = 0$$
$$\downarrow$$
$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ \vdots & \vdots \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix} \begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \vdots \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}$$

Circle Fitting ($Ax=b$)



$$\begin{aligned} x_1^2 - 2x_1 C_x + C_x^2 + y_1^2 - 2y_1 C_y + C_y^2 &= r^2 \\ \vdots & \\ x_n^2 - 2x_n C_x + C_x^2 + y_n^2 - 2y_n C_y + C_y^2 &= r^2 \\ \downarrow & \\ x_i^2 - x_1^2 - 2C_x(x_i - x_1) + y_i^2 - y_1^2 - 2(y_i - y_1)C_y &= 0 \\ \downarrow & \\ \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ \vdots & \vdots \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix} \begin{bmatrix} C_x \\ C_y \end{bmatrix} &= \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \vdots \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix} \end{aligned}$$

Circle Fitting ($Ax=b$)



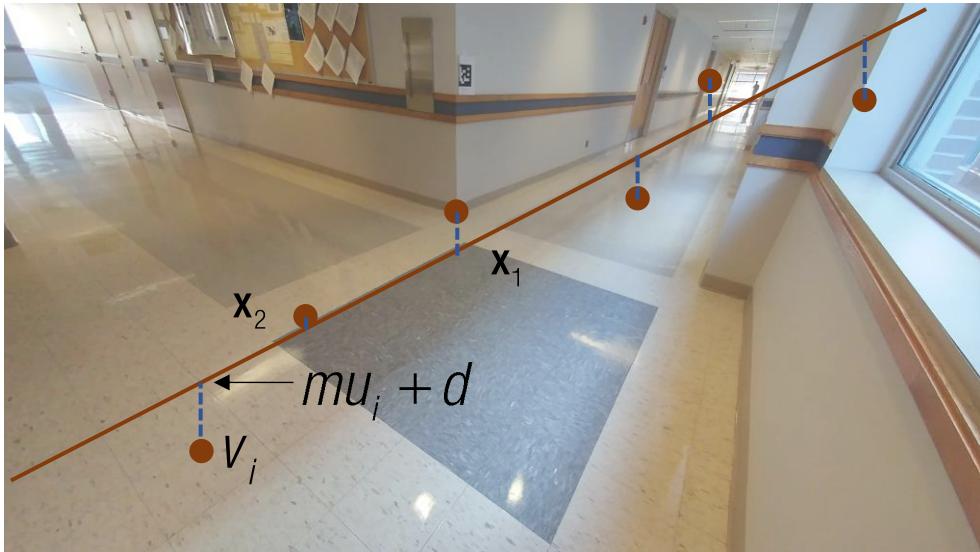
CircleFitting.m

```
cx = 0.3;  
cy = 0.5;  
r = 1;
```

```
theta = 0:0.1:2*pi+0.1;  
theta = theta';  
x = cos(theta) + cx + 0.05*randn(size(theta));  
y = sin(theta) + cy + 0.05*randn(size(theta));
```

```
A = [2*(x(2:end)-x(1)) 2*(y(2:end)-y(1))];  
b = x(2:end).^2-x(1)^2 + y(2:end).^2-y(1)^2;  
u = A\b;  
dist = [x y] - ones(size(x)) * [u(1) u(2)];  
dist = dist';  
dist = mean(sqrt(sum(dist.^2)));
```

Line Fitting ($Ax=b$)



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: $(a, b, c) \longrightarrow (\underline{m}, \underline{d})$

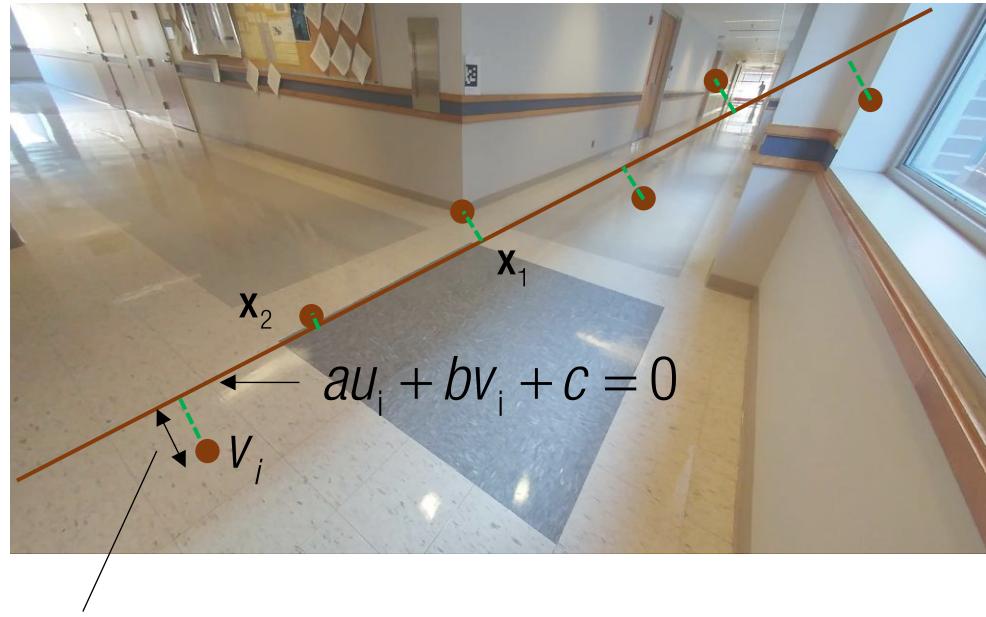
slope y-intercept

$$\begin{array}{l} au_1 + bv_1 + c \approx 0 \\ au_2 + bv_2 + c \approx 0 \\ \vdots \\ au_n + bv_n + c \approx 0 \end{array} \longrightarrow \begin{array}{l} v_1 \approx mu_1 + d \\ v_2 \approx mu_2 + d \\ \vdots \\ v_n \approx mu_n + d \end{array}$$

$$\mathbf{Ax = b}$$

What is different?

Line Fitting ($Ax=0$)



$$d_i = \frac{|au_i + bv_i + c|}{\sqrt{a^2 + b^2}}$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c)

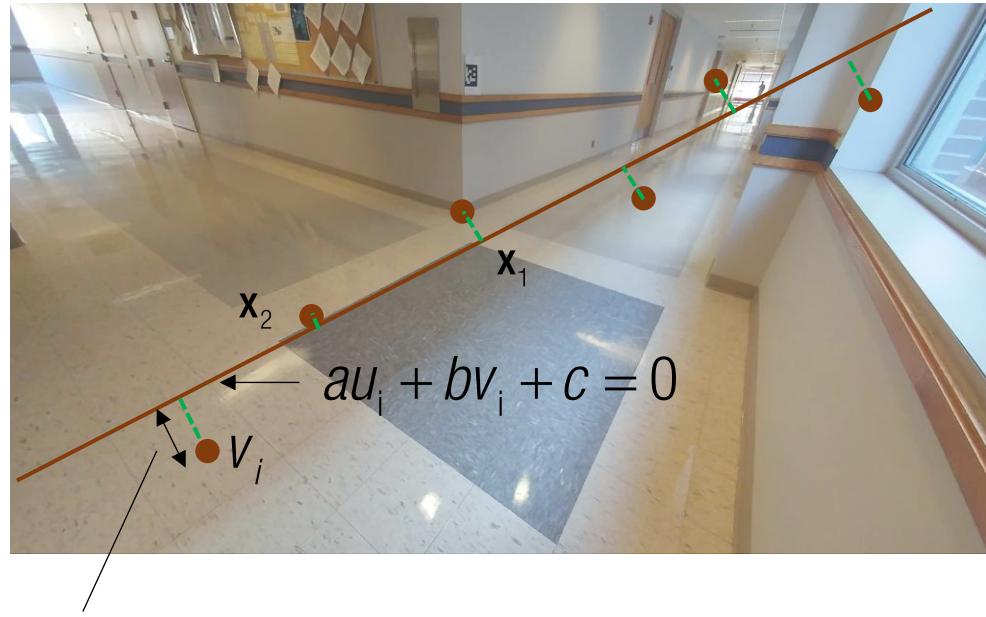
$$\begin{aligned} au_1 + bv_1 + c &\approx 0 \\ au_2 + bv_2 + c &\approx 0 \\ &\vdots \\ au_n + bv_n + c &\approx 0 \end{aligned}$$

$$Ax = 0$$

$$\begin{aligned} v_1 &\approx mu_1 + d \\ v_2 &\approx mu_2 + d \\ &\vdots \\ v_n &\approx mu_n + d \end{aligned}$$

$$Ax = b$$

Line Fitting ($Ax=0$)



$$d_i = \frac{|au_i + bv_i + c|}{\sqrt{a^2 + b^2}}$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

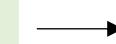
Find the best line: (a, b, c)

$$au_1 + bv_1 + c \approx 0$$

$$au_2 + bv_2 + c \approx 0$$

⋮

$$au_n + bv_n + c \approx 0$$



$$v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

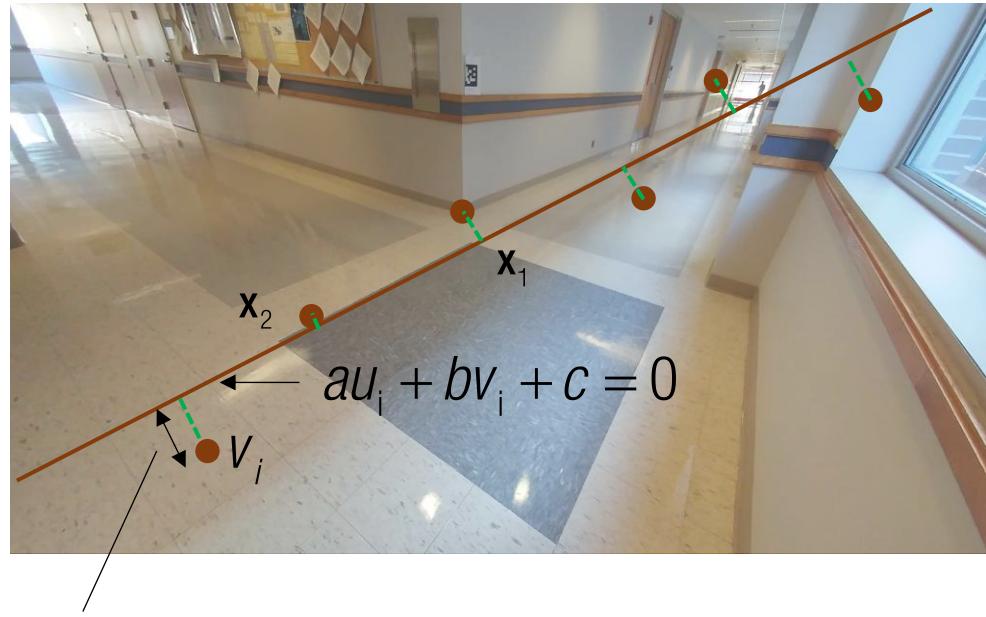
$$v_n \approx mu_n + d$$

$$\mathbf{Ax} = \mathbf{0}$$

$$\mathbf{Ax} = \mathbf{b}$$

Trivial solution: $\mathbf{x} = \mathbf{0}$

Line Fitting ($Ax=0$)



$$d_i = \frac{|au_i + bv_i + c|}{\sqrt{a^2 + b^2}}$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c)

$$\begin{aligned} au_1 + bv_1 + c &\approx 0 \\ au_2 + bv_2 + c &\approx 0 \\ &\vdots \\ au_n + bv_n + c &\approx 0 \end{aligned}$$

$$Ax = 0$$

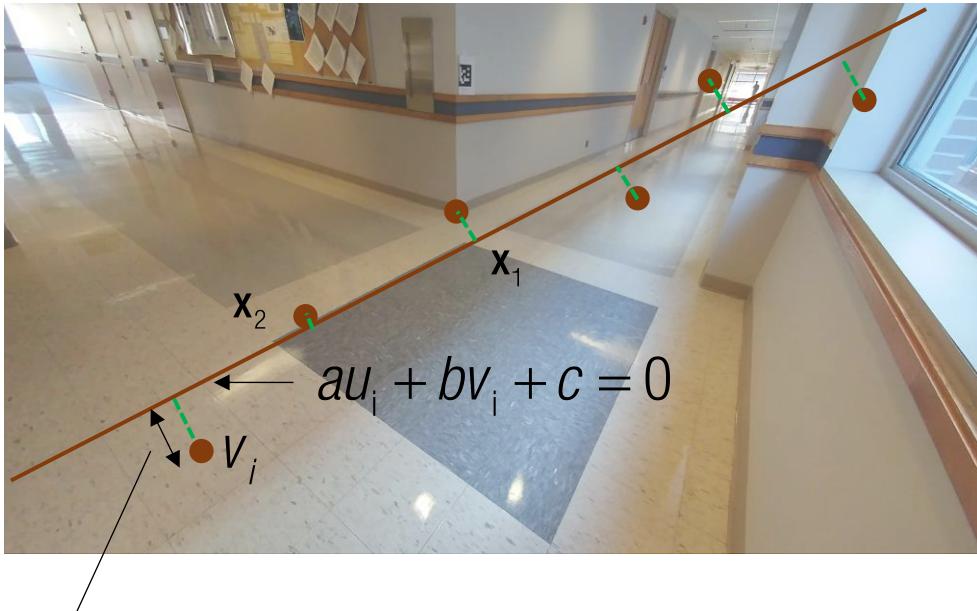
$$\begin{aligned} v_1 &\approx mu_1 + d \\ v_2 &\approx mu_2 + d \\ &\vdots \\ v_n &\approx mu_n + d \end{aligned}$$

$$Ax = b$$

Trivial solution: $\mathbf{x} = \mathbf{0}$

$$\underset{\mathbf{x}}{\text{minimize}} \|Ax\|^2$$

Line Fitting ($\mathbf{Ax} = 0$)



$$d_i = \frac{|au_i + bv_i + c|}{\sqrt{a^2 + b^2}}$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c)

$$\begin{aligned} au_1 + bv_1 + c &\approx 0 \\ au_2 + bv_2 + c &\approx 0 \\ &\vdots \\ au_n + bv_n + c &\approx 0 \end{aligned}$$

$$\mathbf{Ax} = \mathbf{0}$$

$$\begin{aligned} v_1 &\approx mu_1 + d \\ v_2 &\approx mu_2 + d \\ &\vdots \\ v_n &\approx mu_n + d \end{aligned}$$

$$\mathbf{Ax} = \mathbf{b}$$

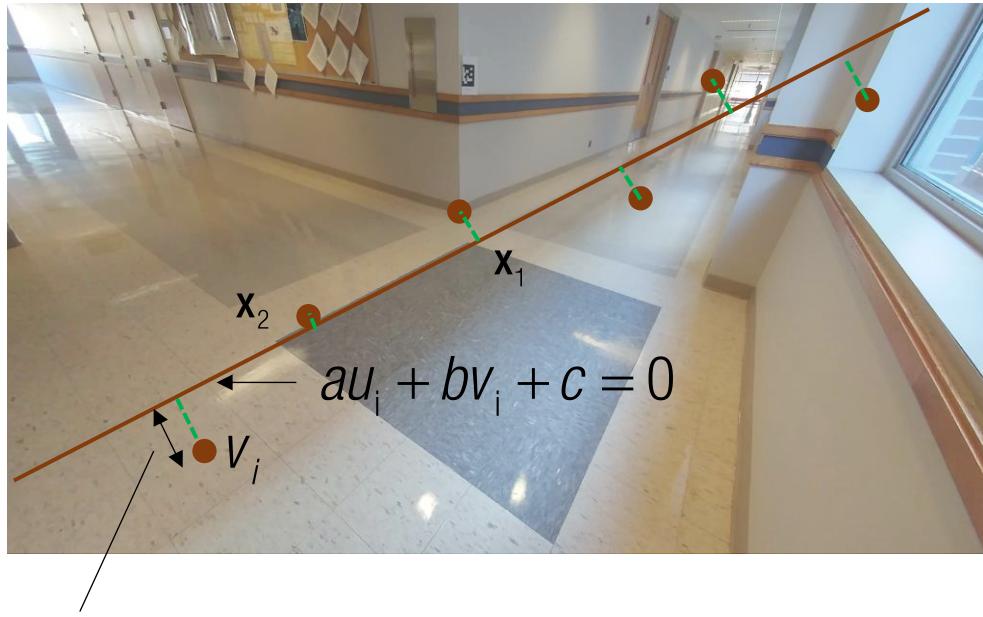
Trivial solution: $\mathbf{x} = \mathbf{0}$

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2$$

$$\text{subject to } \|\mathbf{x}\| = 1$$

Condition to avoid the trivial solution

Line Fitting ($\mathbf{Ax} = 0$)



$$d_i = \frac{|au_i + bv_i + c|}{\sqrt{a^2 + b^2}}$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c)

$$\begin{aligned} au_1 + bv_1 + c &\approx 0 \\ au_2 + bv_2 + c &\approx 0 \\ &\vdots \\ au_n + bv_n + c &\approx 0 \end{aligned}$$

$$\mathbf{Ax} = \mathbf{0}$$

$$\begin{aligned} v_1 &\approx mu_1 + d \\ v_2 &\approx mu_2 + d \\ &\vdots \\ v_n &\approx mu_n + d \end{aligned}$$

$$\mathbf{Ax} = \mathbf{b}$$

Trivial solution: $\mathbf{x} = \mathbf{0}$

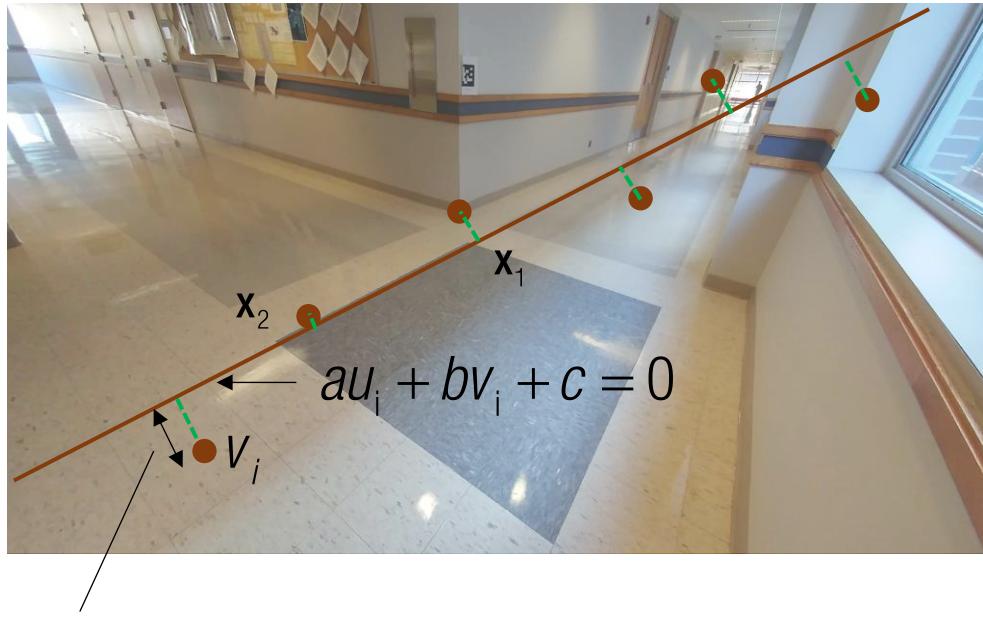
$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2$$

$$\text{subject to } \|\mathbf{x}\| = 1$$

Condition to avoid the trivial solution

How to solve?

Line Fitting ($\mathbf{Ax} = 0$)



$$d_i = \frac{|au_i + bv_i + c|}{\sqrt{a^2 + b^2}}$$

Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c)

$$\begin{aligned} au_1 + bv_1 + c &\approx 0 \\ au_2 + bv_2 + c &\approx 0 \\ &\vdots \\ au_n + bv_n + c &\approx 0 \end{aligned}$$

$$\mathbf{Ax} = \mathbf{0}$$

$$\begin{aligned} v_1 &\approx mu_1 + d \\ v_2 &\approx mu_2 + d \\ &\vdots \\ v_n &\approx mu_n + d \end{aligned}$$

$$\mathbf{Ax} = \mathbf{b}$$

Trivial solution: $\mathbf{x} = \mathbf{0}$

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2 \quad \text{subject to } \|\mathbf{x}\| = 1$$

Condition to avoid the trivial solution

How to solve? approximated null space $\mathbf{x} = \text{null}(\mathbf{A})$

Nullspace

eqs < # unknowns

There exist at least $n-m$ vectors that makes $\mathbf{Ax}=0$.

$$\begin{array}{c|c|c} \mathbf{A} & \mathbf{x} & \mathbf{0} \\ m \times n & n \times 1 & \\ \hline m < n & & \end{array} =$$

$$\longrightarrow \mathbf{x} = \text{null}(\mathbf{A})$$

Singular Value Decomposition (SVD)

eqs < # unknowns

There exist at least $n-m$ vectors that makes $\mathbf{Ax}=0$.

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} \quad \begin{matrix} \mathbf{x} \\ n \times 1 \end{matrix} = \begin{matrix} \mathbf{0} \\ m \times 1 \end{matrix}$$

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} = \begin{matrix} \mathbf{U} \\ m \times m \end{matrix} \begin{matrix} \mathbf{D} \\ m \times n \end{matrix} \begin{matrix} \mathbf{V}^T \\ n \times n \end{matrix}$$

Orthogonal matrix Diagonal matrix Orthogonal matrix

$$\longrightarrow \mathbf{x} = \text{null}(\mathbf{A})$$

Singular Value Decomposition (SVD)

eqs < # unknowns

There exist at least $n-m$ vectors that makes $\mathbf{Ax}=0$.

$$\begin{array}{c} \text{A} \\ \text{m} \times \text{n} \\ \text{m} < \text{n} \end{array} \quad \begin{array}{c} \text{x} \\ \text{n} \times 1 \end{array} = \begin{array}{c} \text{0} \end{array}$$
$$\begin{array}{c} \text{A} \\ \text{m} \times \text{n} \end{array} = \begin{array}{c} \text{U} \\ \text{m} \times \text{m} \end{array} \begin{array}{c} \text{D} \\ \text{m} \times \text{n} \end{array} \begin{array}{c} \text{V}^T \\ \text{n} \times \text{n} \end{array}$$

The diagram illustrates the Singular Value Decomposition (SVD) of a matrix A . The matrix A is shown as a green rectangle labeled $m \times n$. It is decomposed into three components: U , D , and V^T . Matrix U is a light blue rectangle labeled $m \times m$. Matrix D is a red square matrix labeled $m \times n$. Matrix V^T is a yellow rectangle labeled $n \times n$. The total width of U and D is m , and the total width of D and V^T is n . The height of U is m , and the height of V^T is n . The matrix A is shown as a green rectangle labeled $m \times n$. It is equal to the product of U , D , and V^T . The matrix U has a red border. The matrix D is red and has dimensions $m \times n$. The matrix V^T is yellow and has dimensions $n \times n$. Arrows point from the labels to their respective matrices.

$$\longrightarrow \mathbf{x} = \text{null}(\mathbf{A})$$

Singular Value Decomposition (SVD)

eqs < # unknowns

There exist at least $n-m$ vectors that makes $\mathbf{Ax}=0$.

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} \quad \begin{matrix} \mathbf{x} \\ n \times 1 \end{matrix} = \begin{matrix} \mathbf{0} \end{matrix}$$

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} = \begin{matrix} \mathbf{U} \\ m \times m \end{matrix} \begin{matrix} \mathbf{D} \\ m \times n \end{matrix} \begin{matrix} \mathbf{V}^T \\ n \times n \end{matrix}$$

Dimensions: $m \times m$ and $n \times n$

$$\rightarrow \mathbf{x} = \text{null}(\mathbf{A})$$

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} \quad \begin{matrix} \mathbf{V}_{m+1:n} \\ n \times n \end{matrix} = \begin{matrix} \mathbf{0} \end{matrix} \quad \mathbf{V}_{m+1:n} = \text{null}(\mathbf{A})$$

Singular Value Decomposition (SVD)

eqs < # unknowns

There exist at least $n-m$ vectors that makes $\mathbf{Ax}=0$.

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} \quad \begin{matrix} \mathbf{x} \\ n \times 1 \end{matrix} = \begin{matrix} \mathbf{0} \end{matrix}$$

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} = \begin{matrix} \mathbf{U} \\ m \times m \end{matrix} \begin{matrix} \mathbf{D} \\ m \times n \end{matrix} \begin{matrix} \mathbf{V}^T \\ n \times n \end{matrix}$$

Dimensions: $m \times m$, $m \times n-m$, $n-m \times n$.
Arrows indicate dimensions: m (vertical), $n-m$ (horizontal).

$$\rightarrow \mathbf{x} = \text{null}(\mathbf{A})$$

$$\begin{matrix} \mathbf{A} \\ \mathbf{V}_{:, \text{end}} \end{matrix} = \begin{matrix} \mathbf{0} \end{matrix} \quad \mathbf{V}_{:, \text{end}} = \text{null}(\mathbf{A})$$

For a unique solution, \mathbf{A} should be $m \times (m+1)$

Singular Value Decomposition (SVD)

eqs > # unknowns

There exist no nullspace of \mathbf{A} .

$$\begin{array}{c|c|c} \mathbf{A} & \mathbf{x} & \approx \mathbf{0} \\ m \times n & n \times 1 & \\ \hline m > n \end{array}$$

$$\begin{array}{c|c|c|c} \mathbf{A} & = & \mathbf{U} & \mathbf{V}^T \\ m \times n & & m \times n & n \times n \\ & & \mathbf{D} & \\ & & n \times n & n \times n \end{array}$$

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2 \text{ subject to } \|\mathbf{x}\| = 1$$

Singular Value Decomposition (SVD)

eqs > # unknowns

There exist no nullspace of \mathbf{A} .

$$\begin{array}{c|c|c} \mathbf{A} & \mathbf{x} & \approx \mathbf{0} \\ m \times n & n \times 1 & \\ \hline & m > n & \end{array}$$

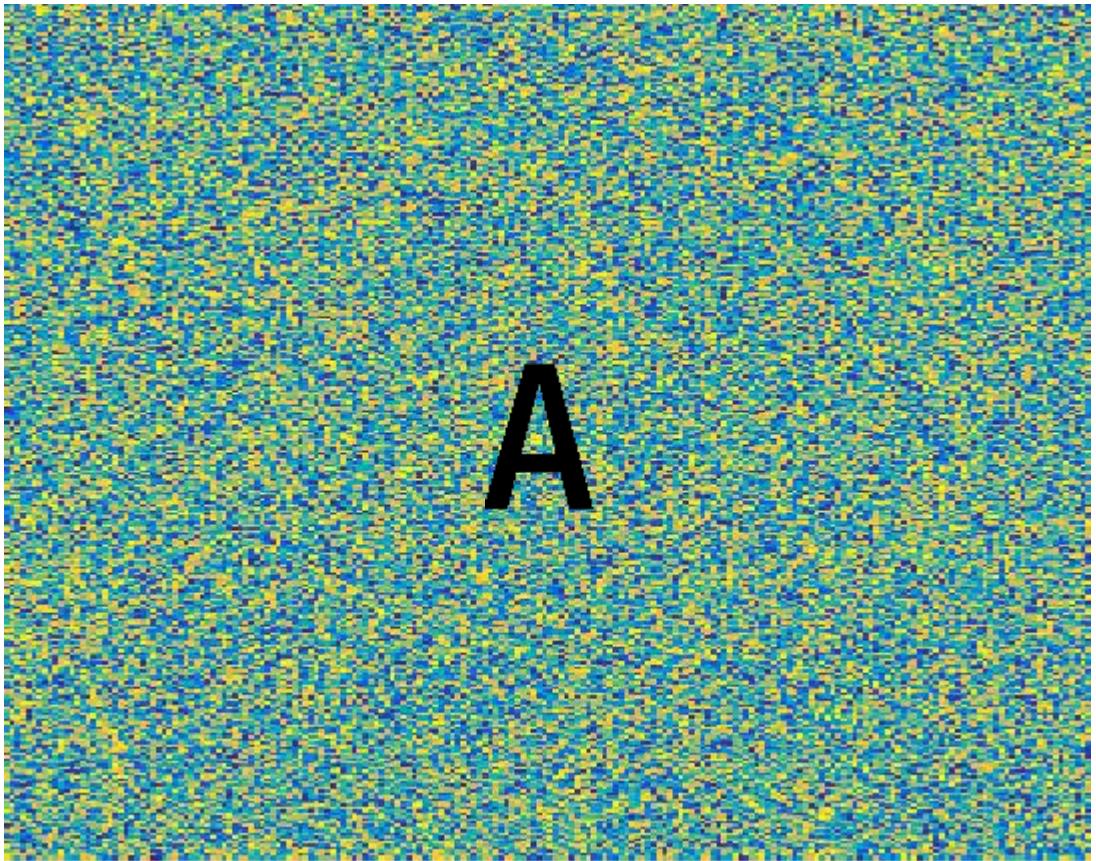
$$\begin{array}{c|c|c|c} \mathbf{A} & = & \mathbf{U} & \mathbf{V}^T \\ m \times n & & m \times n & n \times n \\ & & \mathbf{D} & \\ & & n \times n & n \times n \\ \hline & & \leftarrow \text{Last row} & \end{array}$$

$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2 \text{ subject to } \|\mathbf{x}\| = 1$$

Approximated nullspace of \mathbf{A} :

$$\mathbf{V}_{:, \text{end}}$$

Random Matrix SVD

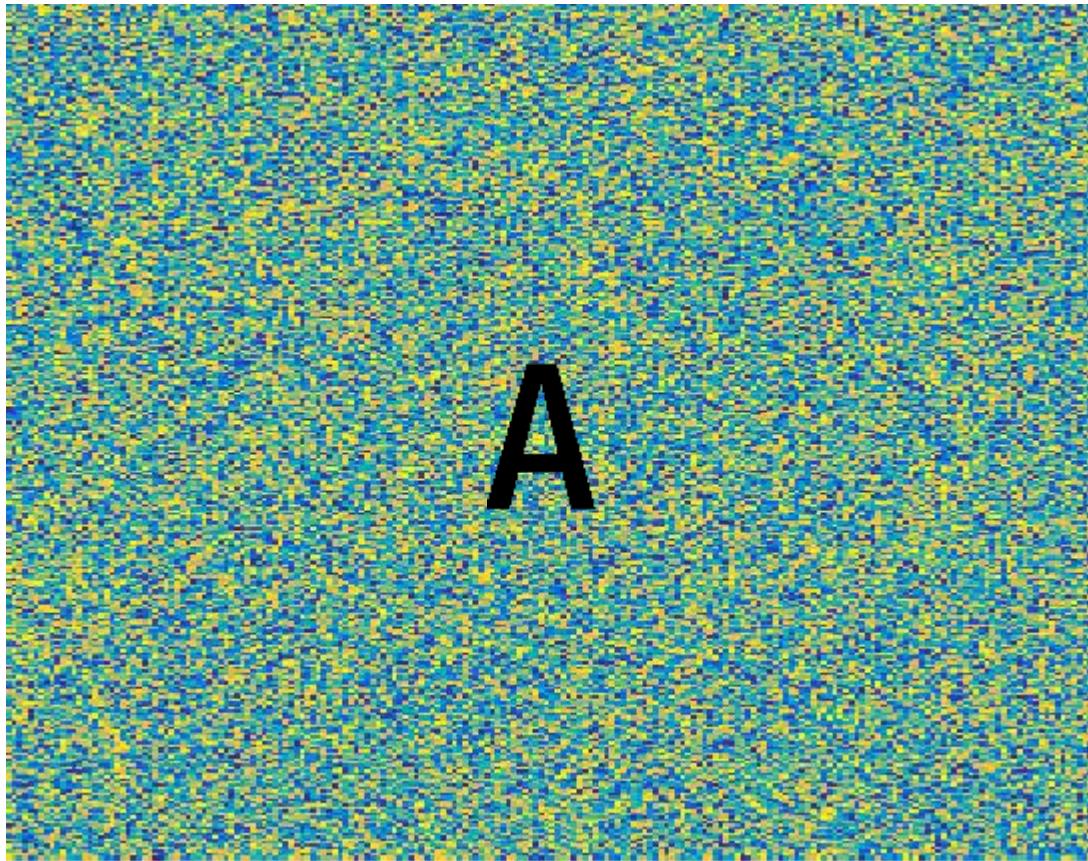


Random matrix

$$= \begin{matrix} U & \begin{matrix} D \\ V^T \end{matrix} \end{matrix}$$

The matrix A is shown to be equal to its Singular Value Decomposition (SVD) components. The decomposition is represented as $A = U \begin{pmatrix} D \\ V^T \end{pmatrix}$. The matrix U is a light blue rectangular block. The matrix D is a 4x4 diagonal matrix with red entries, enclosed in a red border. The matrix V^T is a yellow rectangular block.

Residual (Nullspace Approximation)

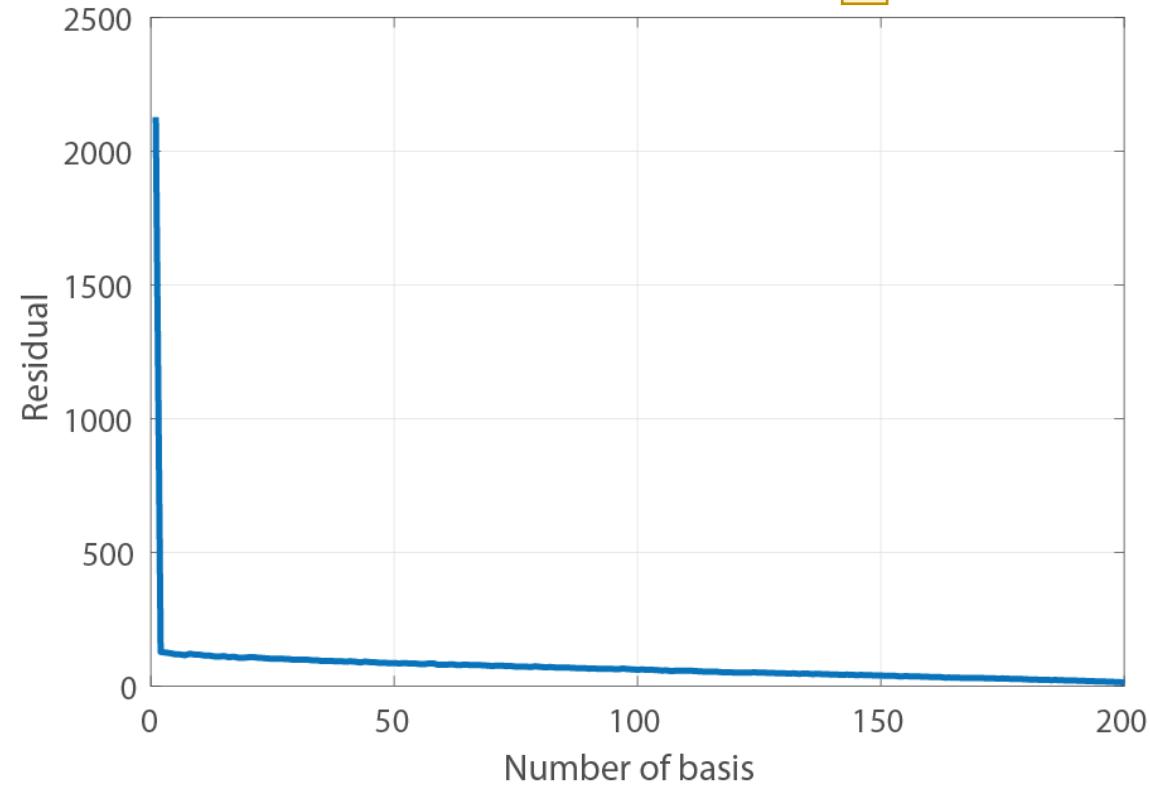


Random matrix

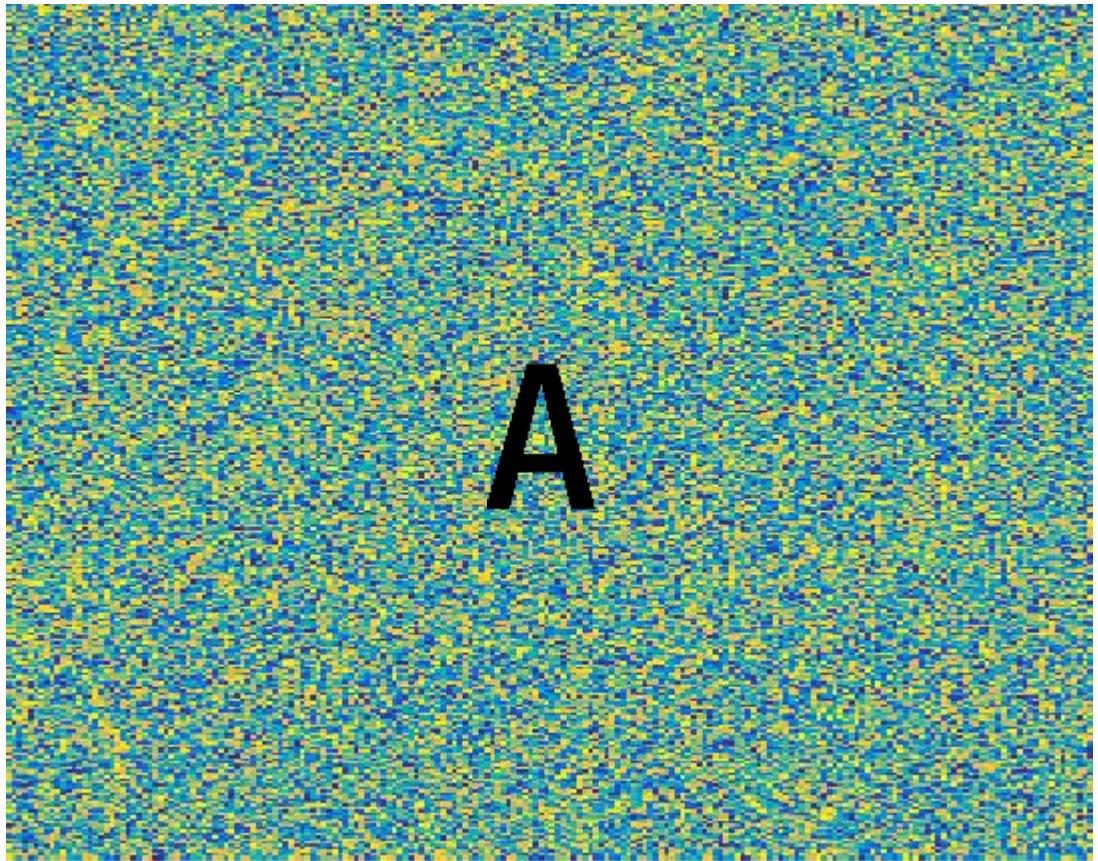
$\mathbf{V}_{:,i}$

Approximated nullspace of \mathbf{A} :

$\mathbf{V}_{:,end}$



SVD Matrix Approximation

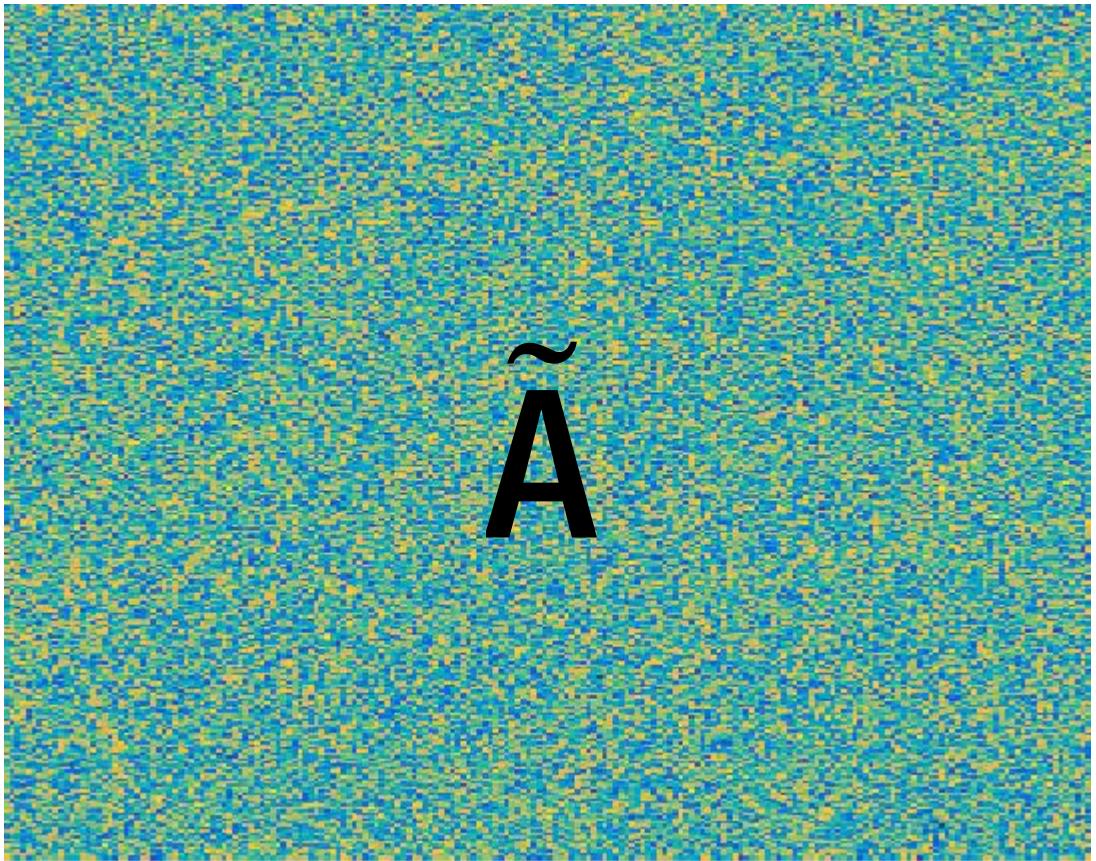


Random matrix

$$= \begin{matrix} U & D \\ V^T & \end{matrix}$$

The matrix **A** is shown as a sum of three matrices: **U**, **D**, and **V^T**. **U** is a 10x10 matrix with a light blue background and a dark blue diagonal line. **D** is a 10x10 matrix with a dark purple background and a single light blue diagonal line. **V^T** is a 10x10 matrix with a light green background and a dark green diagonal line.

SVD Matrix Approximation

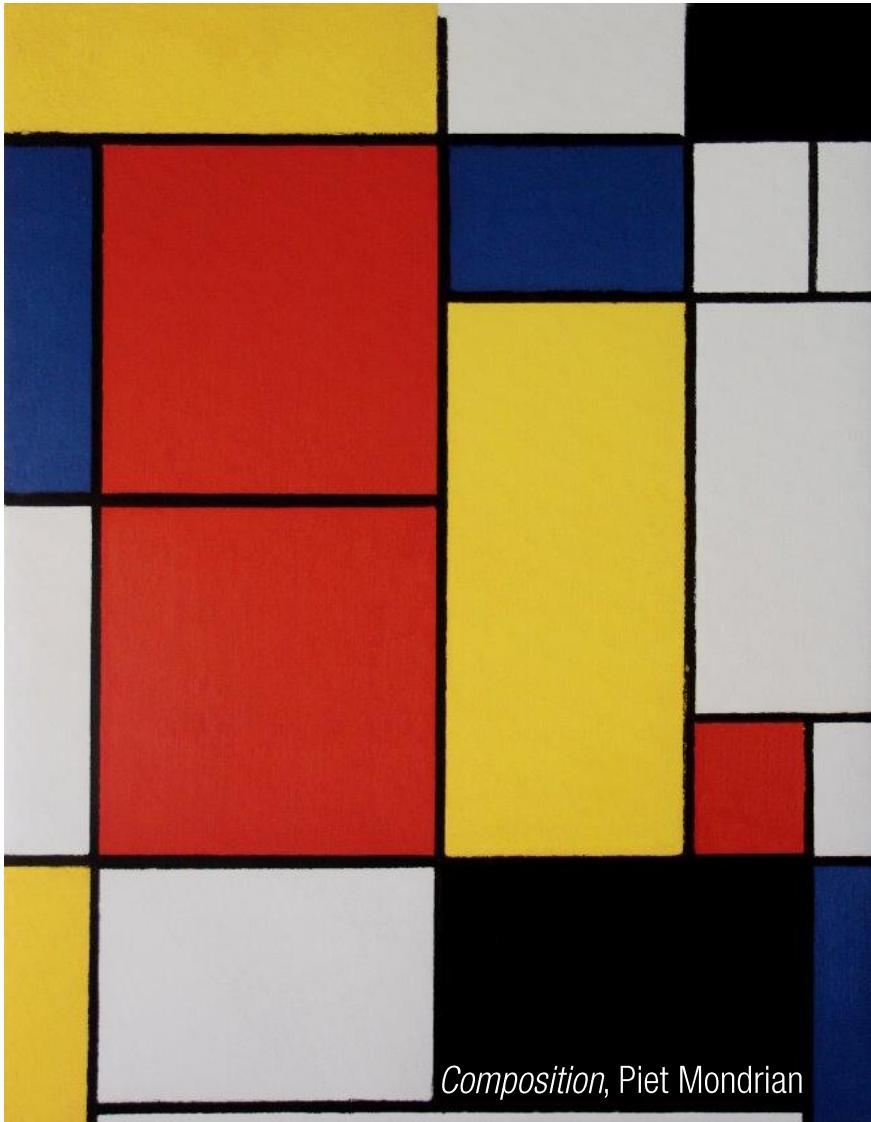


Reconstructed matrix

$$= \mathbf{U} \mathbf{D} \mathbf{V}^T$$

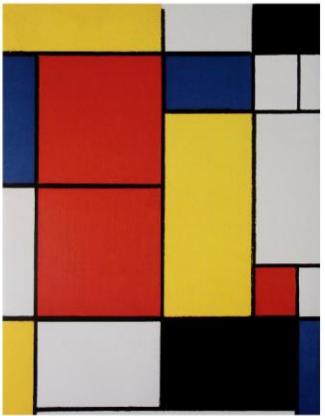
The equation shows the decomposition of the matrix $\tilde{\mathbf{A}}$ into three matrices: \mathbf{U} , \mathbf{D} , and \mathbf{V}^T .
- \mathbf{U} is a blue rectangular matrix.
- \mathbf{D} is a diagonal matrix with dark blue values along the main diagonal.
- \mathbf{V}^T is a light blue rectangular matrix.

Mondrian Painting SVD

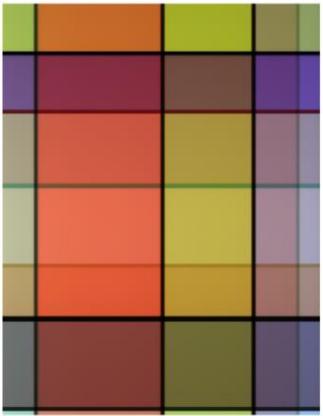


$$= \begin{matrix} U & D & V^T \\ m \times n & n \times n & n \times n \end{matrix}$$

Mondrian Painting SVD Approximation



Ground truth



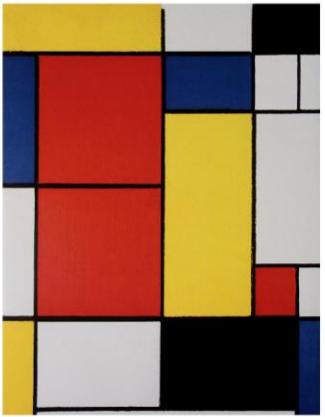
Number of basis: 1

MondrianSVD.m

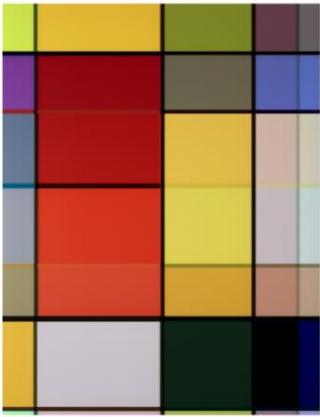


$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

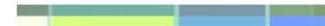
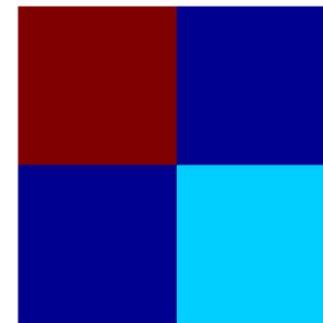


Ground truth



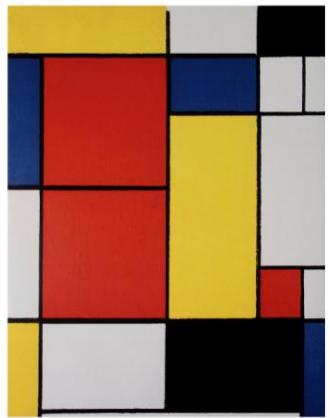
Number of basis: 2

MondrianSVD.m

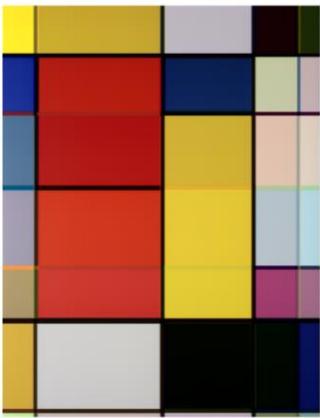


$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

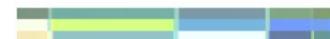
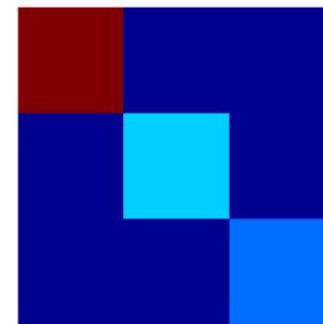


Ground truth



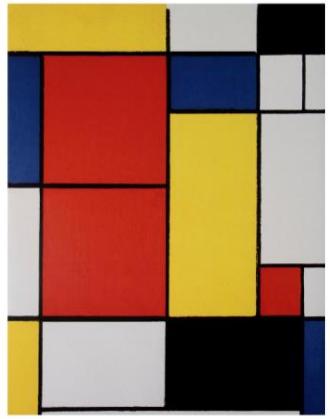
Number of basis: 3

MondrianSVD.m

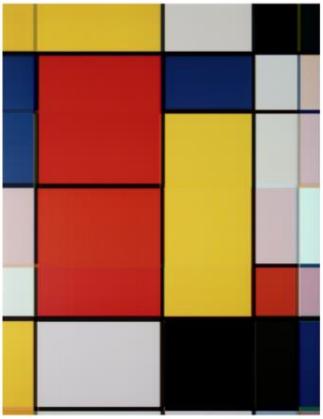


$$A = U D V^T$$

Mondrian Painting SVD Approximation

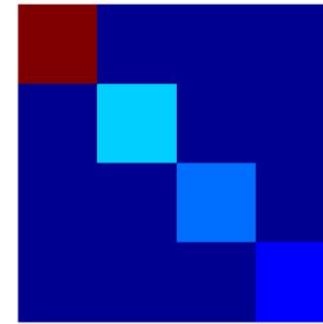


Ground truth



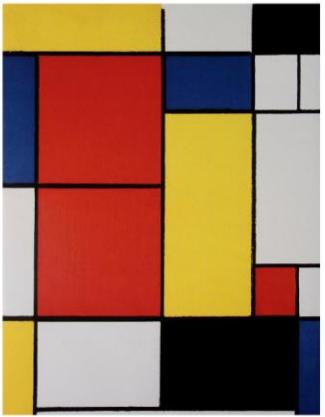
Number of basis: 4

MondrianSVD.m

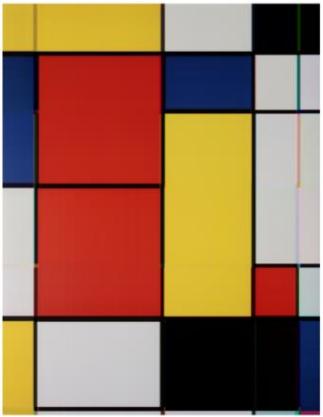


$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

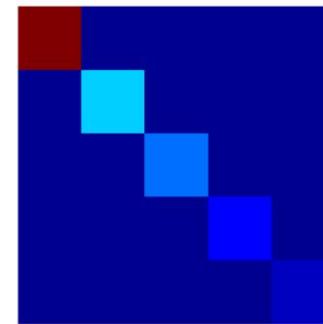


Ground truth



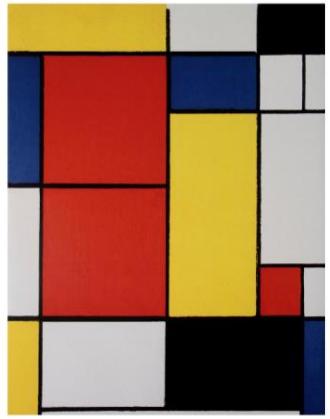
Number of basis: 5

MondrianSVD.m

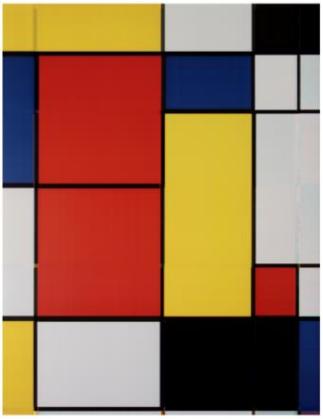


$$A = U D V^T$$

Mondrian Painting SVD Approximation

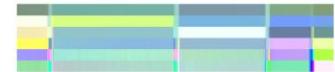
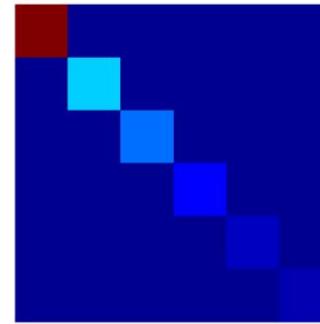
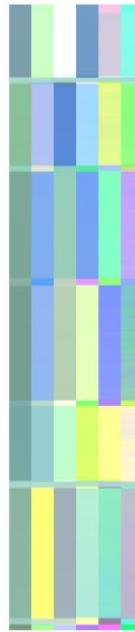


Ground truth



Number of basis: 6

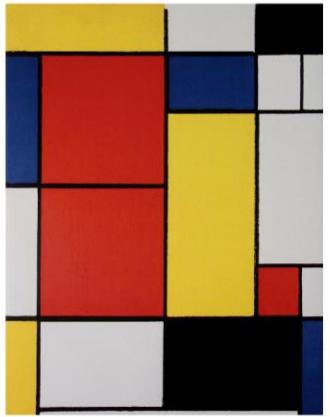
MondrianSVD.m



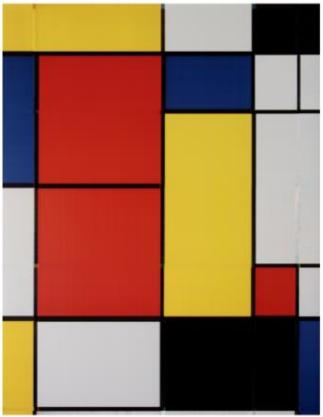
$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

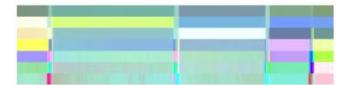
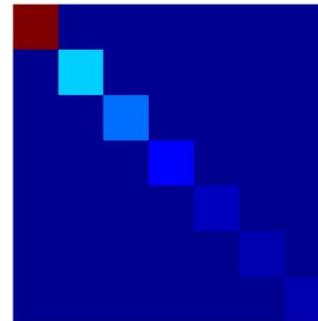
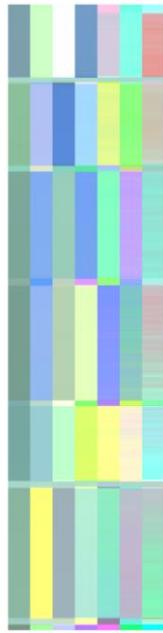
MondrianSVD.m



Ground truth

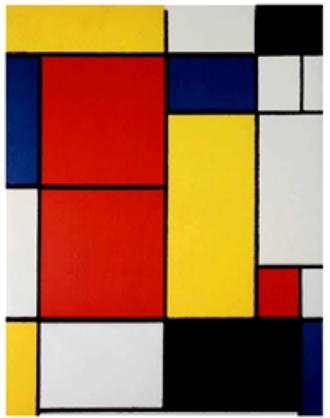


Number of basis: 7

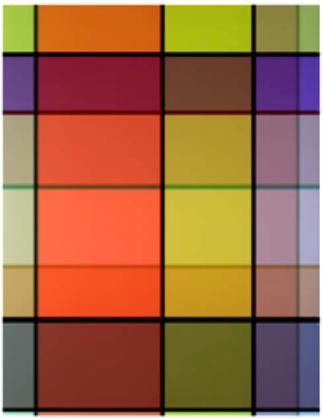


$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

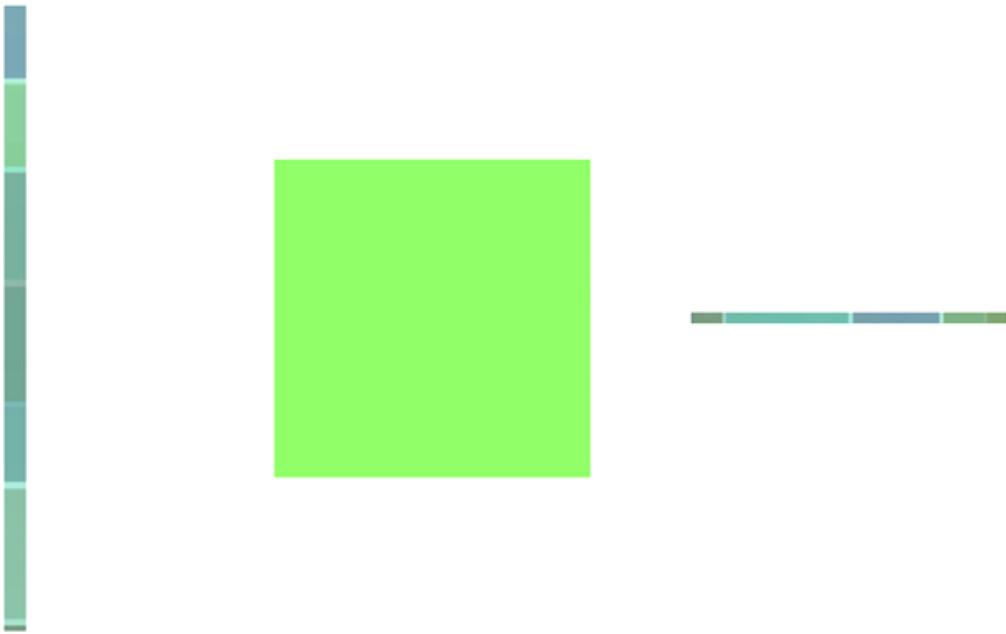


Ground truth



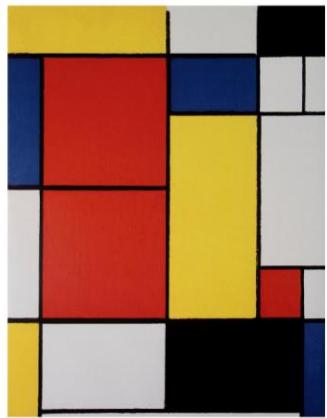
Number of basis: 1

MondrianSVD.m

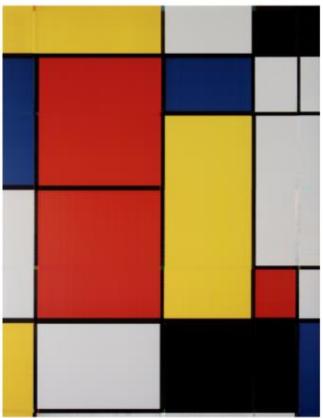


$$A = U D V^T$$

Reconstruction Error



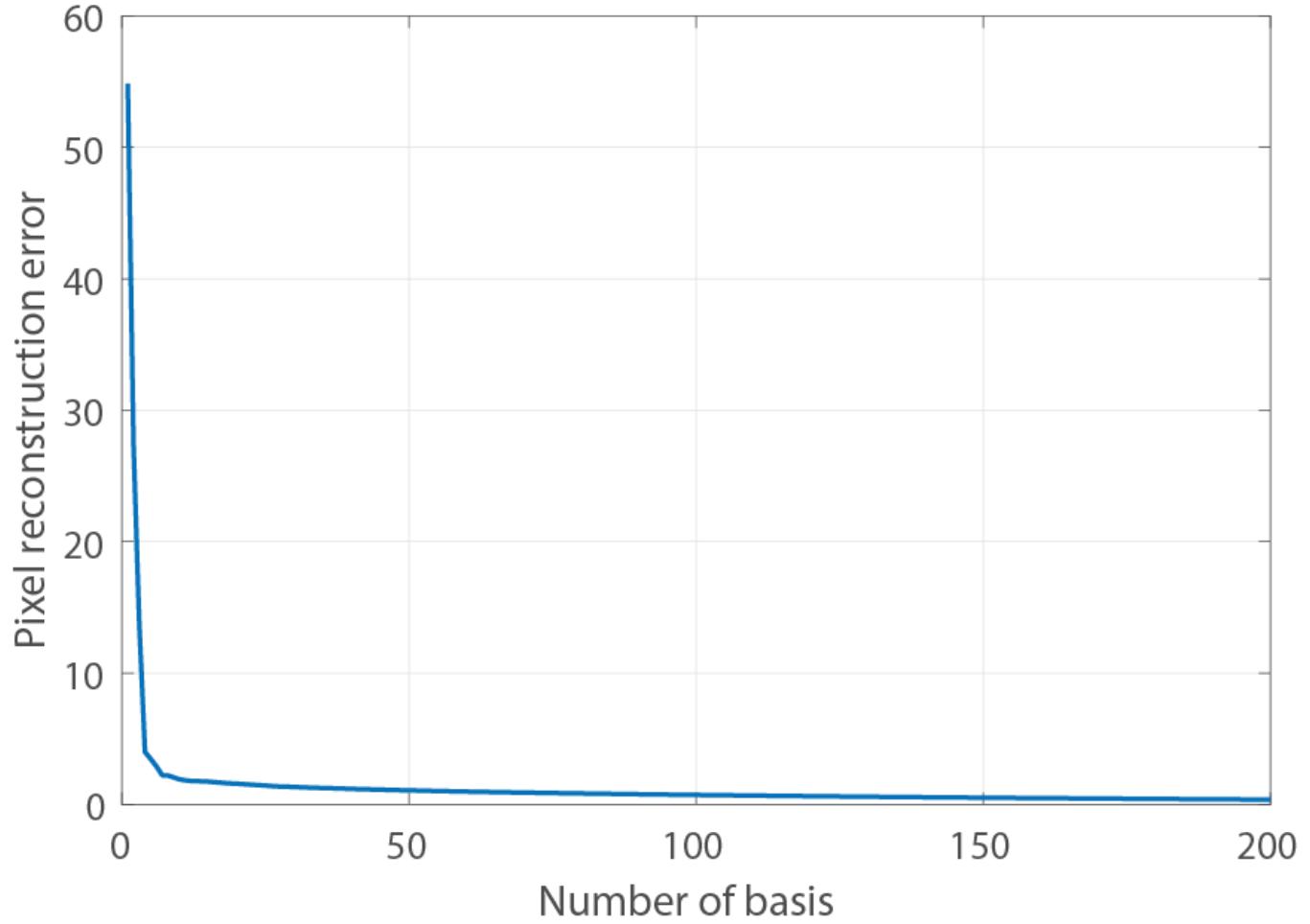
Ground truth



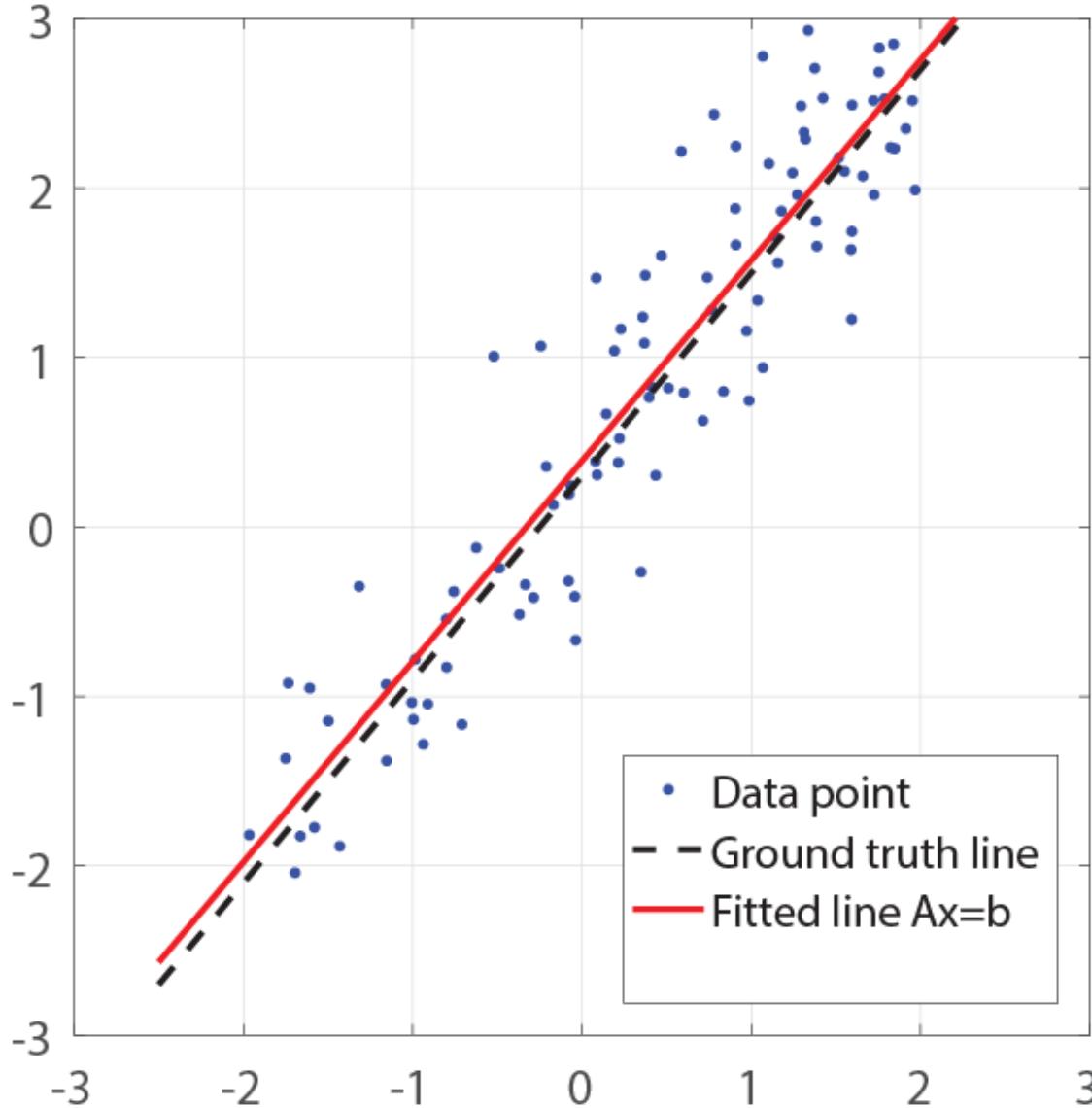
Number of basis: 7

A

MondrianSVD.m



Line Fitting ($Ax=b$)

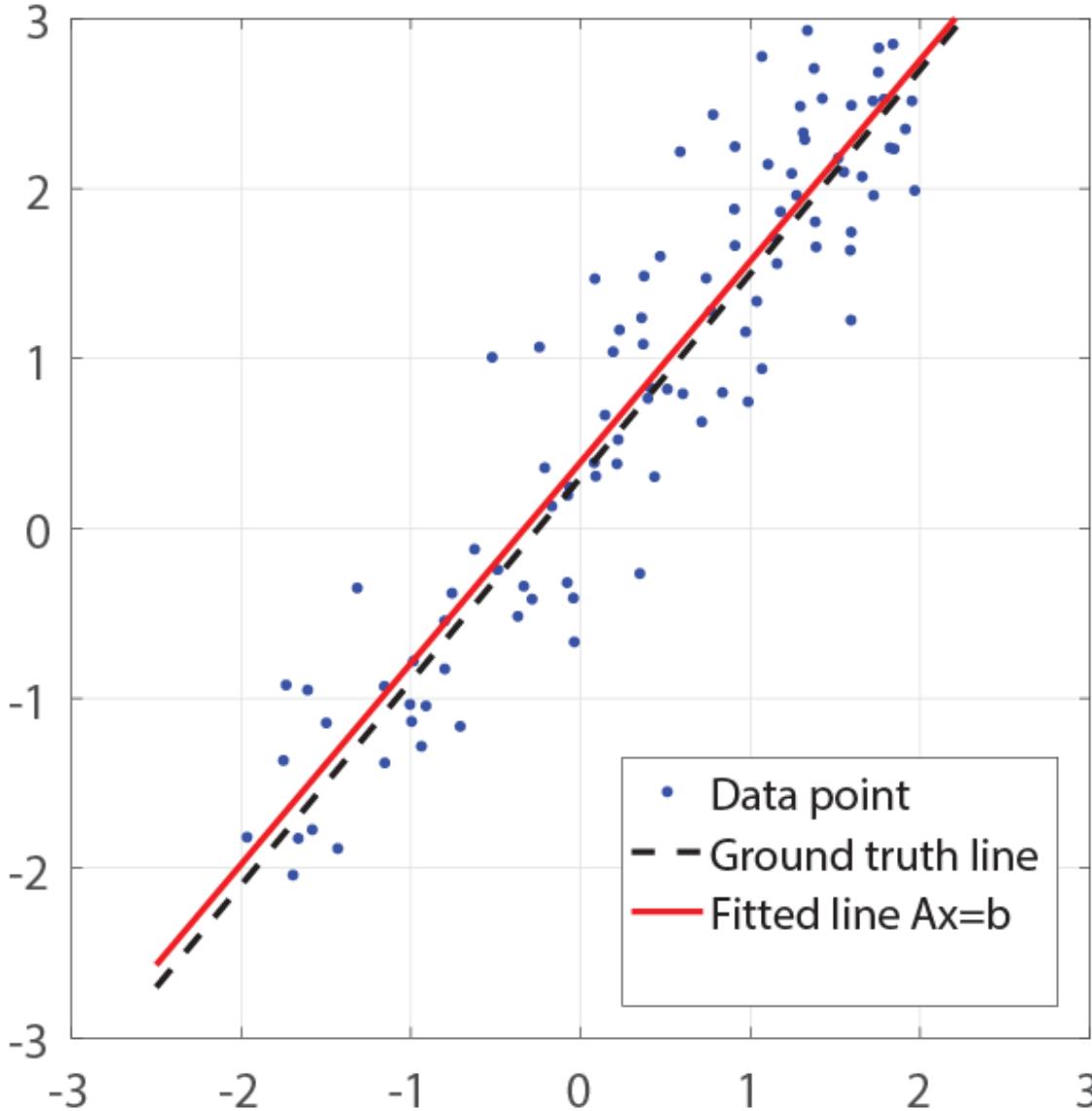


Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$
Find the best line: (m, d)

$$\begin{aligned}v_1 &\approx mu_1 + d \\v_2 &\approx mu_2 + d \\&\vdots \\v_n &\approx mu_n + d\end{aligned}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c) (m, d)

$$au_1 + bv_1 + c \approx 0$$

$$au_2 + bv_2 + c \approx 0$$

⋮

$$au_n + bv_n + c \approx 0$$

$$v_1 \approx mu_1 + d$$

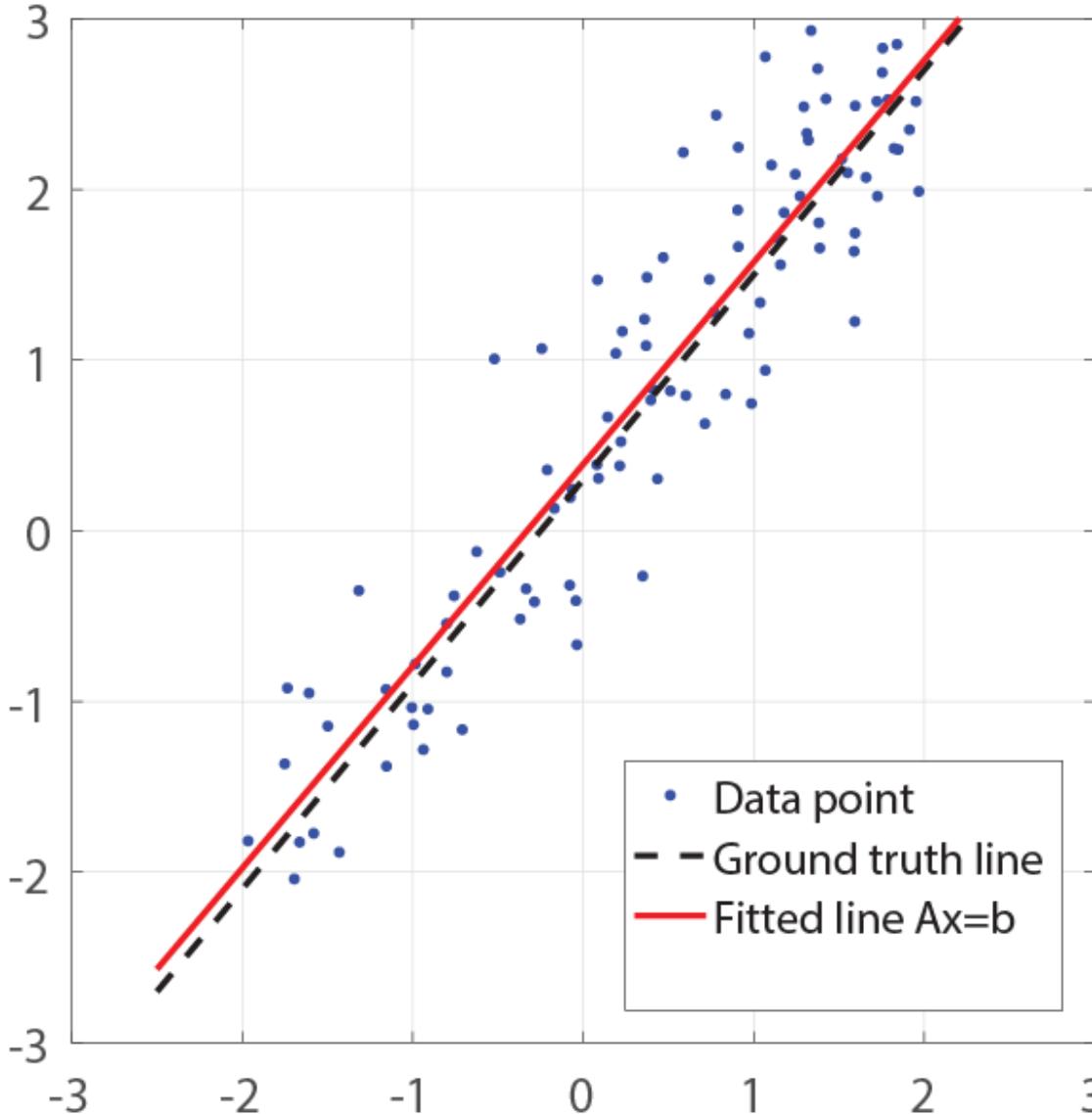
$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c) (m, d)

$$au_1 + bv_1 + c \approx 0$$

$$au_2 + bv_2 + c \approx 0$$

⋮

$$au_n + bv_n + c \approx 0$$

$$v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

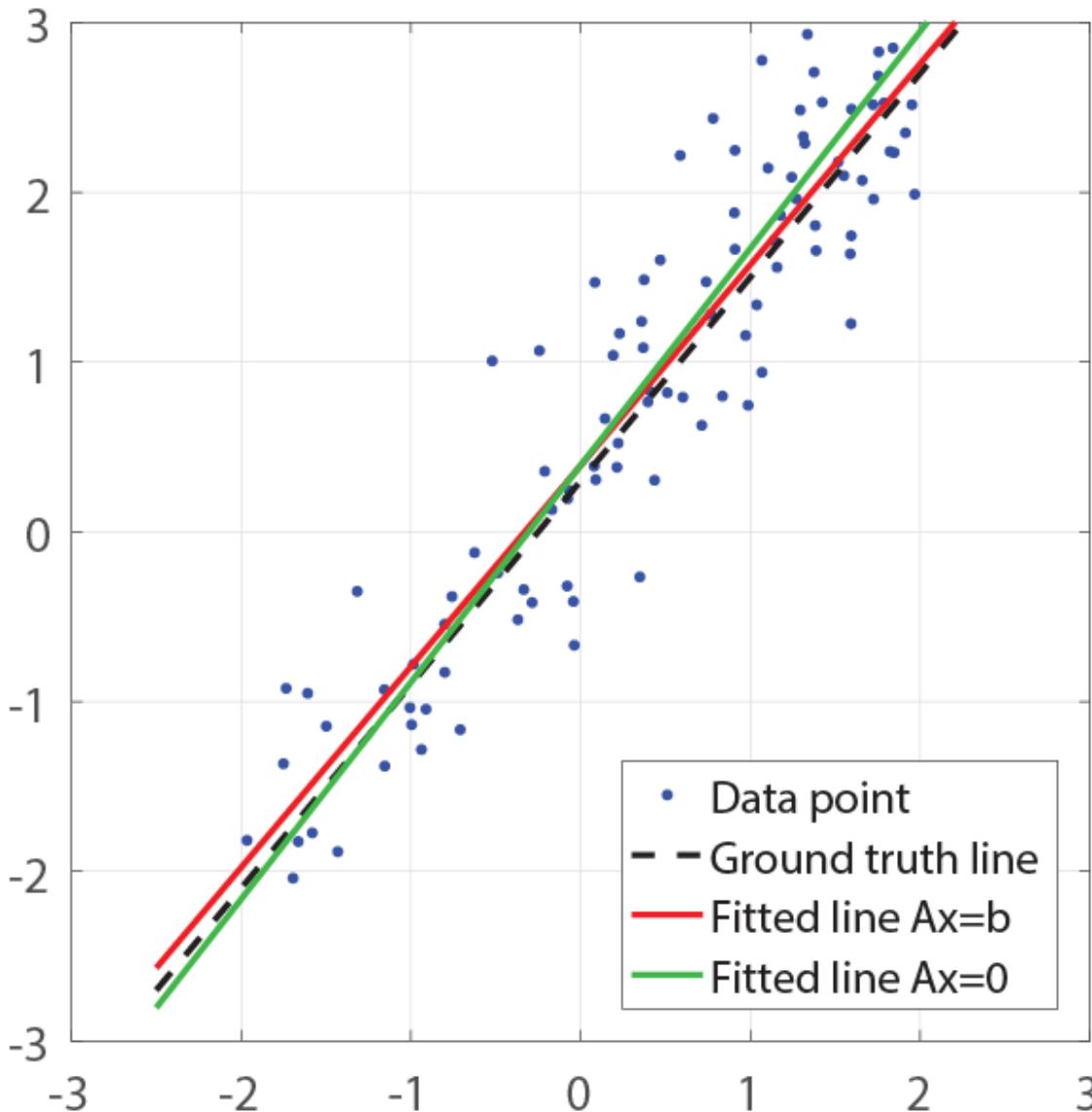
⋮

$$v_n \approx mu_n + d$$

$$\begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ \vdots & \vdots & 1 \\ u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Line Fitting ($Ax=b$)



Given points: $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$

Find the best line: (a, b, c)

$$au_1 + bv_1 + c \approx 0$$

$$au_2 + bv_2 + c \approx 0$$

⋮

$$au_n + bv_n + c \approx 0$$



$$v_1 \approx mu_1 + d$$

$$v_2 \approx mu_2 + d$$

⋮

$$v_n \approx mu_n + d$$

$$\begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots \\ u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$



How to compute homography?

Homography Computation



$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Homography Computation



$$v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Homography Computation



$$v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$\begin{aligned} \rightarrow & h_{11}u_x + h_{12}u_y + h_{13} + h_{31}u_x v_x + h_{32}u_y v_x + h_{33}v_x = 0 \\ & h_{21}u_x + h_{22}u_y + h_{23} + h_{31}u_x v_y + h_{32}u_y v_y + h_{33}v_y = 0 \end{aligned}$$

$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Unknowns: h_{11}, \dots, h_{33}

Equations: 2 per correspondence

Homography Computation



$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

$$v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$\rightarrow \begin{aligned} h_{11}u_x + h_{12}u_y + h_{13} - h_{31}u_xv_x - h_{32}u_yv_x - h_{33}v_x &= 0 \\ h_{21}u_x + h_{22}u_y + h_{23} - h_{31}u_xv_y - h_{32}u_yv_y - h_{33}v_y &= 0 \end{aligned}$$

$$\rightarrow \begin{bmatrix} u_x & u_y & 1 & -u_xv_x & -u_yv_x & -v_x \\ & & u_x & u_y & 1 & -u_xv_y & -u_yv_y & -v_y \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Homography Computation



$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

$$v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

$$\rightarrow \begin{aligned} h_{11}u_x + h_{12}u_y + h_{13} - h_{31}u_xv_x - h_{32}u_yv_x - h_{33}v_x &= 0 \\ h_{21}u_x + h_{22}u_y + h_{23} - h_{31}u_xv_y - h_{32}u_yv_y - h_{33}v_y &= 0 \end{aligned}$$

$$\rightarrow \begin{bmatrix} u_x & u_y & 1 & -u_xv_x & -u_yv_x & -v_x \\ & & u_x & u_y & -u_xv_y & -u_yv_y & -v_y \end{bmatrix} \begin{matrix} \mathbf{A} \\ 2 \times 9 \end{matrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Recall: Singular Value Decomposition (SVD)

eqs < # unknowns

There exist at least $n-m$ vectors that makes $\mathbf{Ax}=0$.

$$\begin{matrix} \mathbf{A} \\ m \times n \end{matrix} \quad \begin{matrix} \mathbf{x} \\ n \times 1 \end{matrix} = \begin{matrix} \mathbf{0} \end{matrix}$$

$$\longrightarrow \mathbf{x} = \text{null}(\mathbf{A})$$

$$\begin{matrix} \mathbf{A} \\ m \times (m+1) \end{matrix} \quad \begin{matrix} \mathbf{v}_{:,end} \\ m+1 \times 1 \end{matrix} = \begin{matrix} \mathbf{0} \end{matrix} \quad \mathbf{v}_{:,end} = \text{null}(\mathbf{A})$$

For a unique solution, \mathbf{A} should be $m \times (m+1)$

Homography Computation

How many correspondences are needed?



$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u_x & u_y & 1 \\ u_x & u_y & 1 \end{bmatrix} \begin{bmatrix} -u_x v_x & -u_y v_x & -v_x \\ -u_x v_y & -u_y v_y & -v_y \end{bmatrix} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

2x9

Homography Computation

How many correspondences are needed? 4



$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u_x & u_y & 1 \\ u_x & u_y & 1 \end{bmatrix} \mathbf{A} \begin{bmatrix} -u_x v_x & -u_y v_x & -v_x \\ -u_x v_y & -u_y v_y & -v_y \end{bmatrix} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

8x9

$$\mathbf{x} = \mathbf{V}_{:,end}^T = \text{null}(\mathbf{A})$$

Homography Computation

How many correspondences are needed? 4



$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

ComputeHomography.m

```
function H = ComputeHomography(u, X)
```

```
A = [];
for i = 1 : size(u,1)
    A = [A; X(i,:)-u(i,1)*X(i,:)];
    A = [A; zeros(1,3)-u(i,2)*X(i,:)];
end
```

Constructing A

Homography Computation

How many correspondences are needed? 4



$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

ComputeHomography.m

```
function H = ComputeHomography(u, X)
```

```
A = [];
for i = 1 : size(u,1)
    A = [A; X(i,:) zeros(1,3) -u(i,1)*X(i,:)];
    A = [A; zeros(1,3) X(i,:) -u(i,2)*X(i,:)];
end
```

```
[u, d, v] = svd(A);
h = v(:,end);
H = [h(1:3)'; h(4:6)'; h(7:9)'];
H = H/norm(H);
```

Constructing **A**

Solving **Ax=0**

KellerEntranceHomography.m

```
im1 = imread('keller_left.png');
im2 = imread('keller_right.png');

im_warped = zeros(2000,4000,3);

u1 = [2806 1004;    2456 753;
      1677 1234;    2325 1474];

u2 = [1483 1541;    1948 997;
      860 843;    587 1316];

u1 = [u1 ones(4,1)];
u2 = [u2 ones(4,1)];

H1 = ComputeHomography(u2, u1);
H2 = ComputeHomography(u1, u2);

im_warped1 = ImageWarping(im1, H1);
im_warped2 = ImageWarping(im2, H2);

im_1 = 0.5*im_warped1 + 0.5*im2;
im_2 = 0.5*im_warped2 + 0.5*im1;
```

