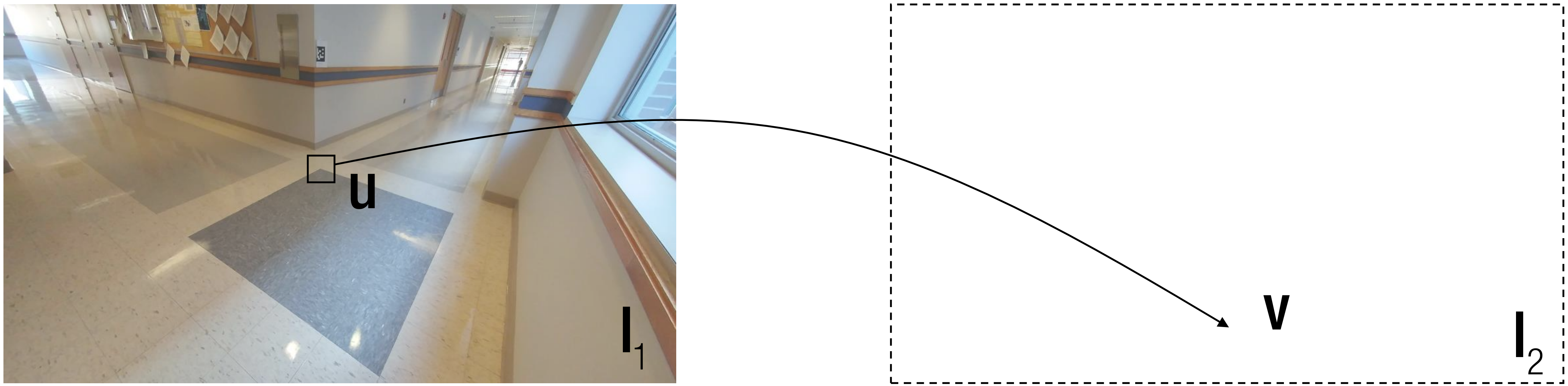
Image Transform

Panorama Image (Keller+Lind Hall)

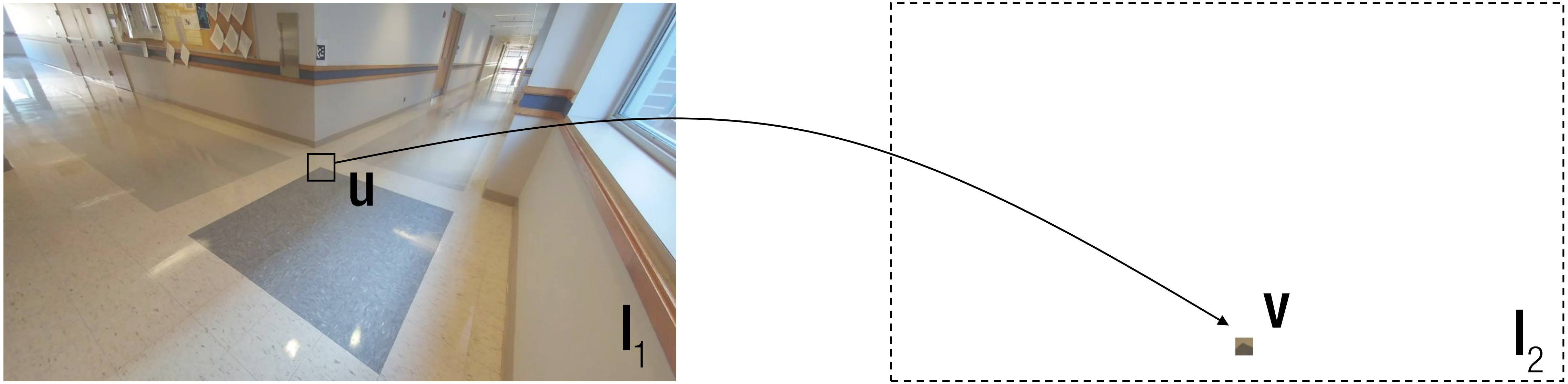
Francois Vogel

Image Warping (Coordinate Transform)



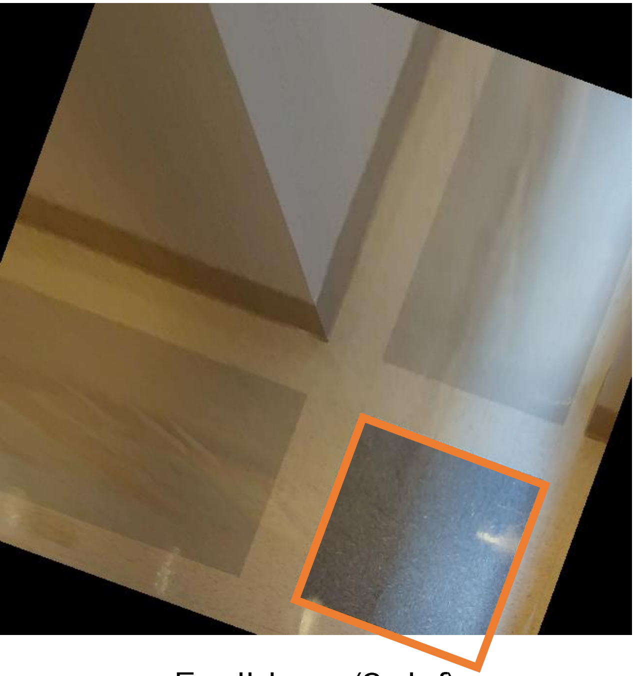
$$\mathbf{I}_2(\mathbf{v}) = \mathbf{I}_1(\mathbf{u})$$

Image Warping (Coordinate Transform)



$$\mathbf{I}_2(\mathbf{v}) = \mathbf{I}_1(\mathbf{u})$$: Pixel transport

Hierarchy of Transformations



Euclidean (3 dof)

- Length
- Angle
- Area

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

Similarity (4 dof)

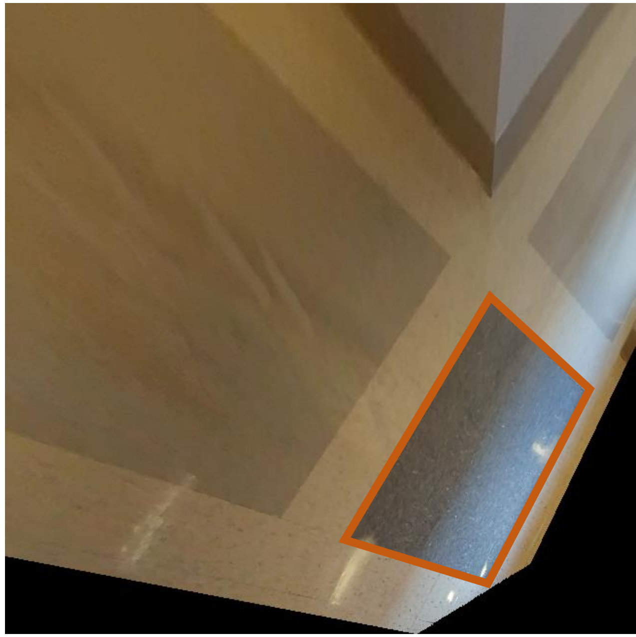
- Length ratio
- Angle

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

Affine (6 dof)

- Parallelism
- Ratio of area
- Ratio of length

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

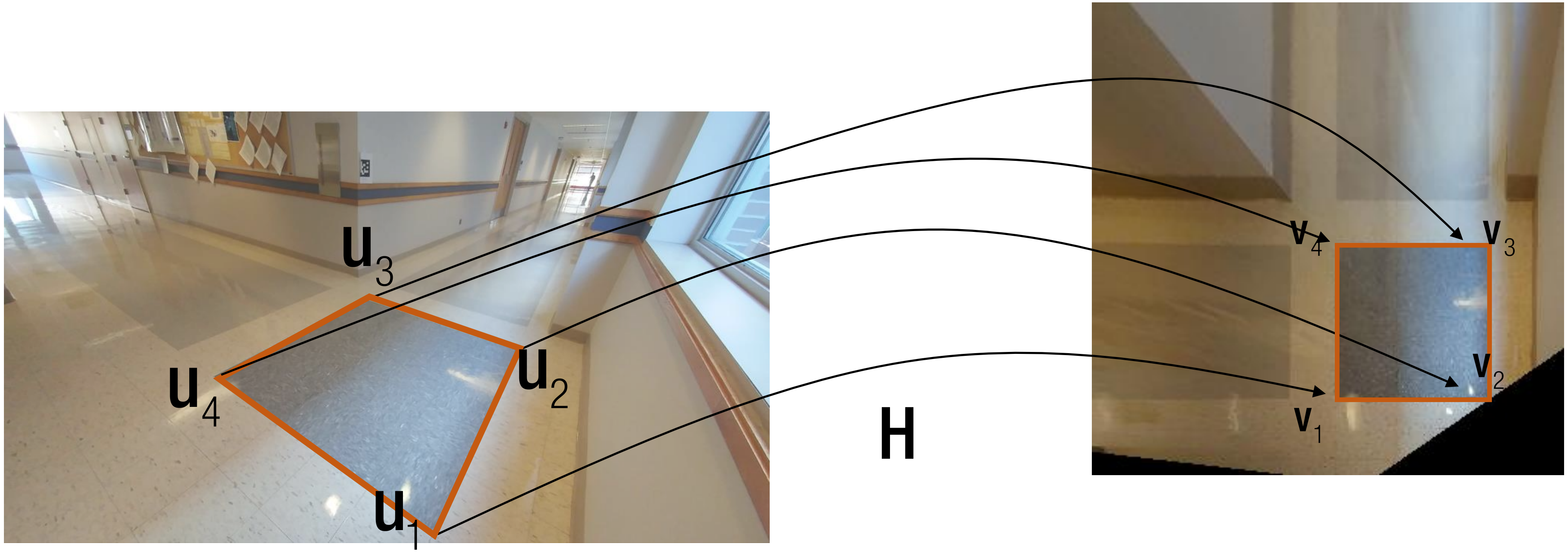


Projective (8 dof)

- Cross ratio
- Concurrency
- Colinearity

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

Fun with Homography



Fun with Homography

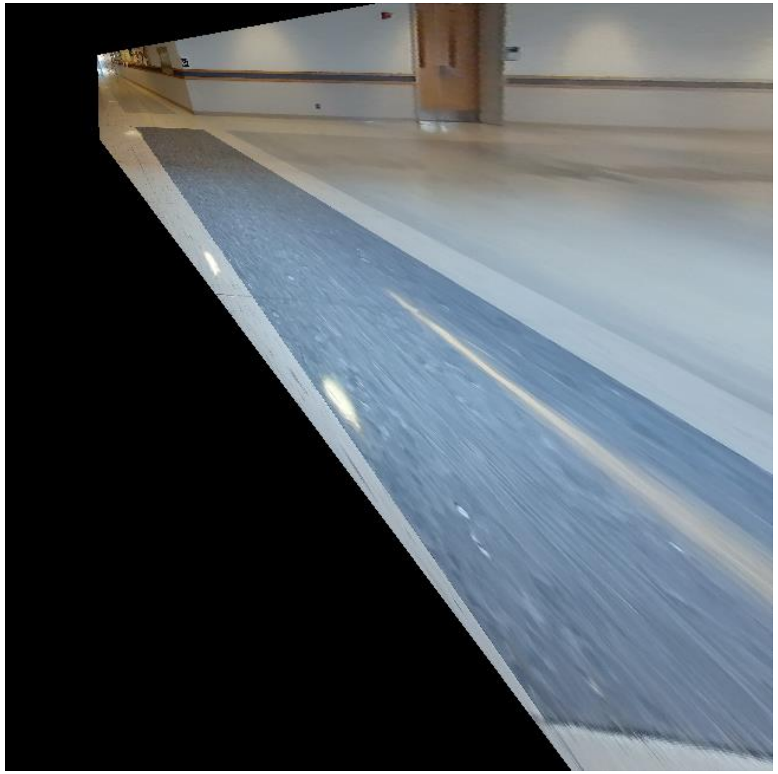


Image Inpainting

Image Inpainting

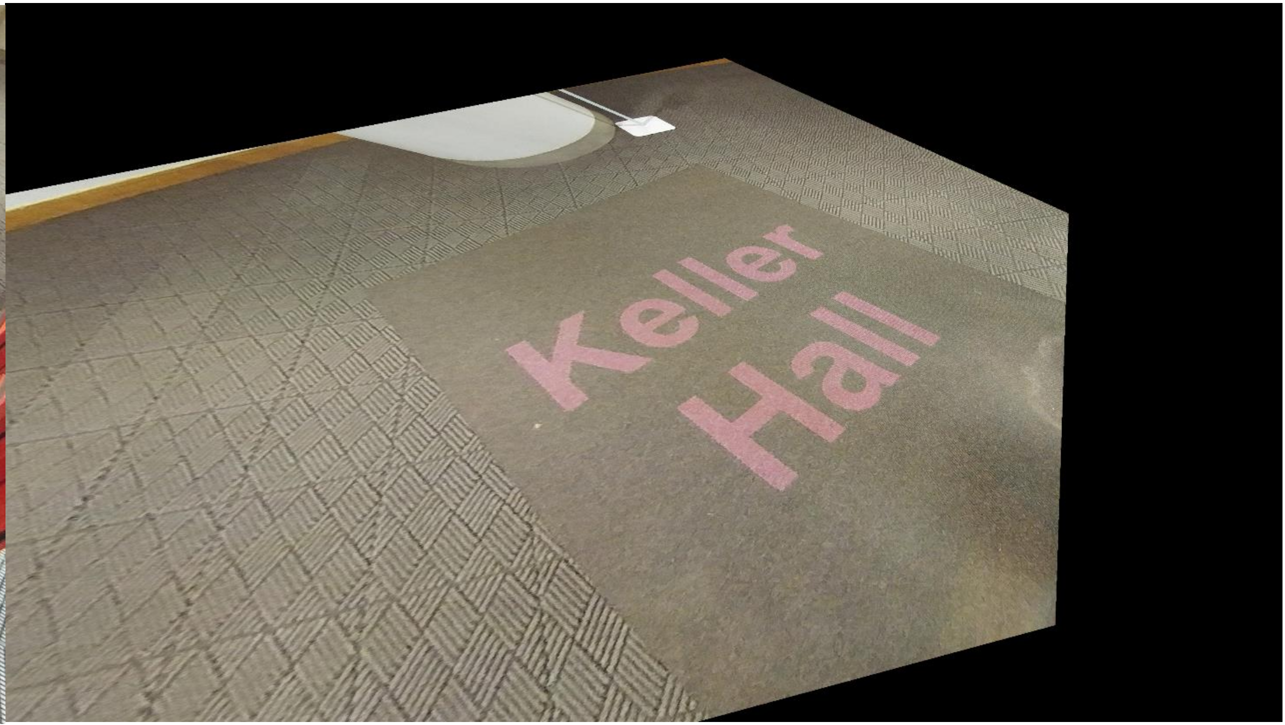
Image Transform via 3D Plane

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

360 Panorama

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

Image Transform by Pure 3D Rotation

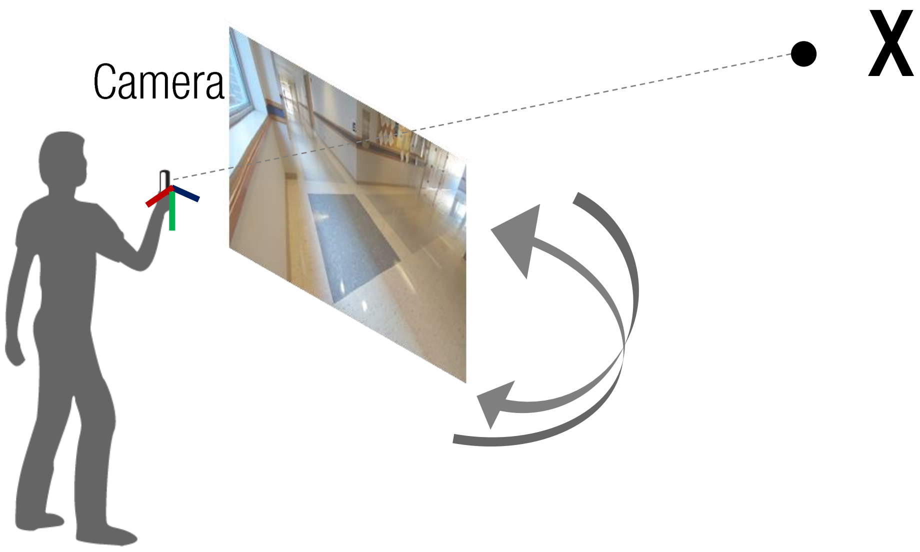
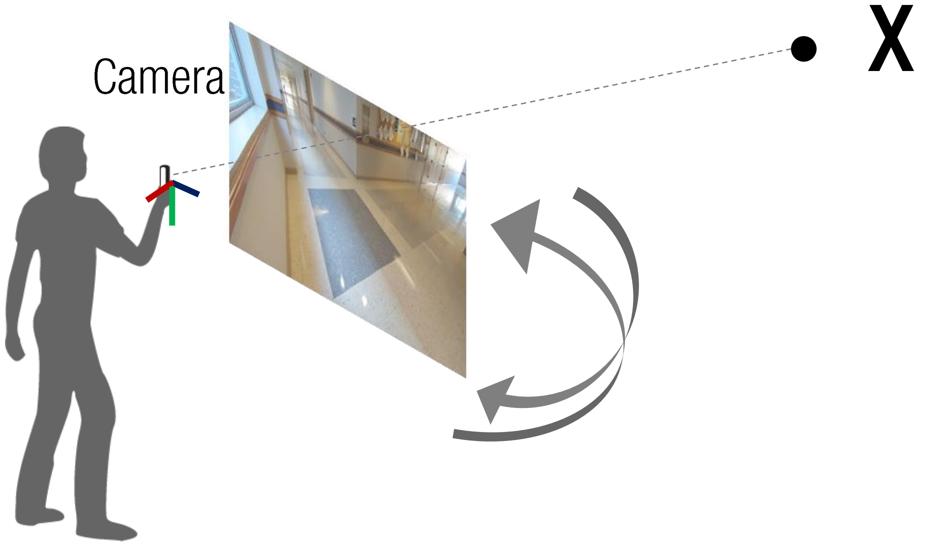


Image Transform by Pure 3D Rotation



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

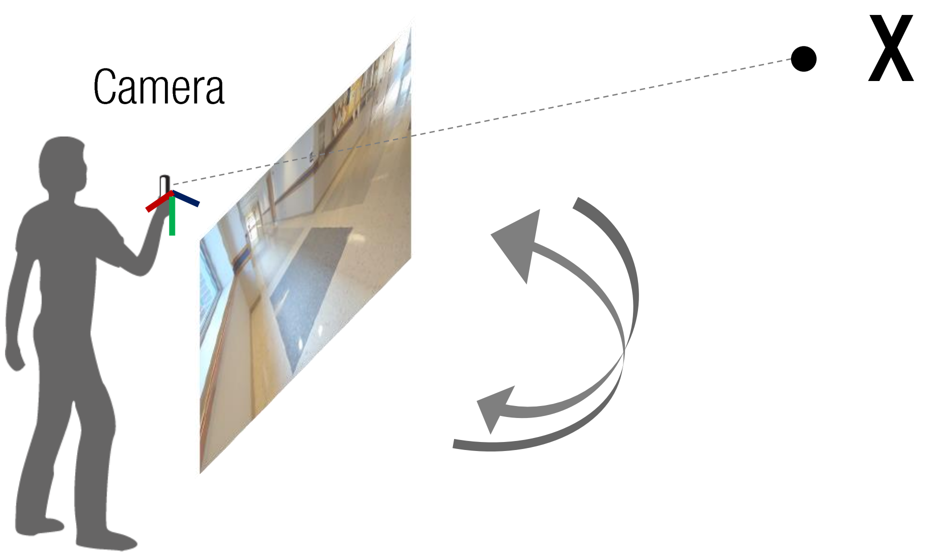
Image Transform by Pure 3D Rotation

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

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

$$\mathbf{t} = 0$$

Image Transform by Pure 3D Rotation



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

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

$$\mathbf{t} = 0$$

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

Image Transform by Pure 3D Rotation

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

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

$$\mathbf{t} = 0$$

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

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

Image Transform by Pure 3D Rotation

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

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

$$\mathbf{t} = 0$$

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

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

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

Image Transform by Pure 3D Rotation

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

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

$$\mathbf{t} = 0$$

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

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

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

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

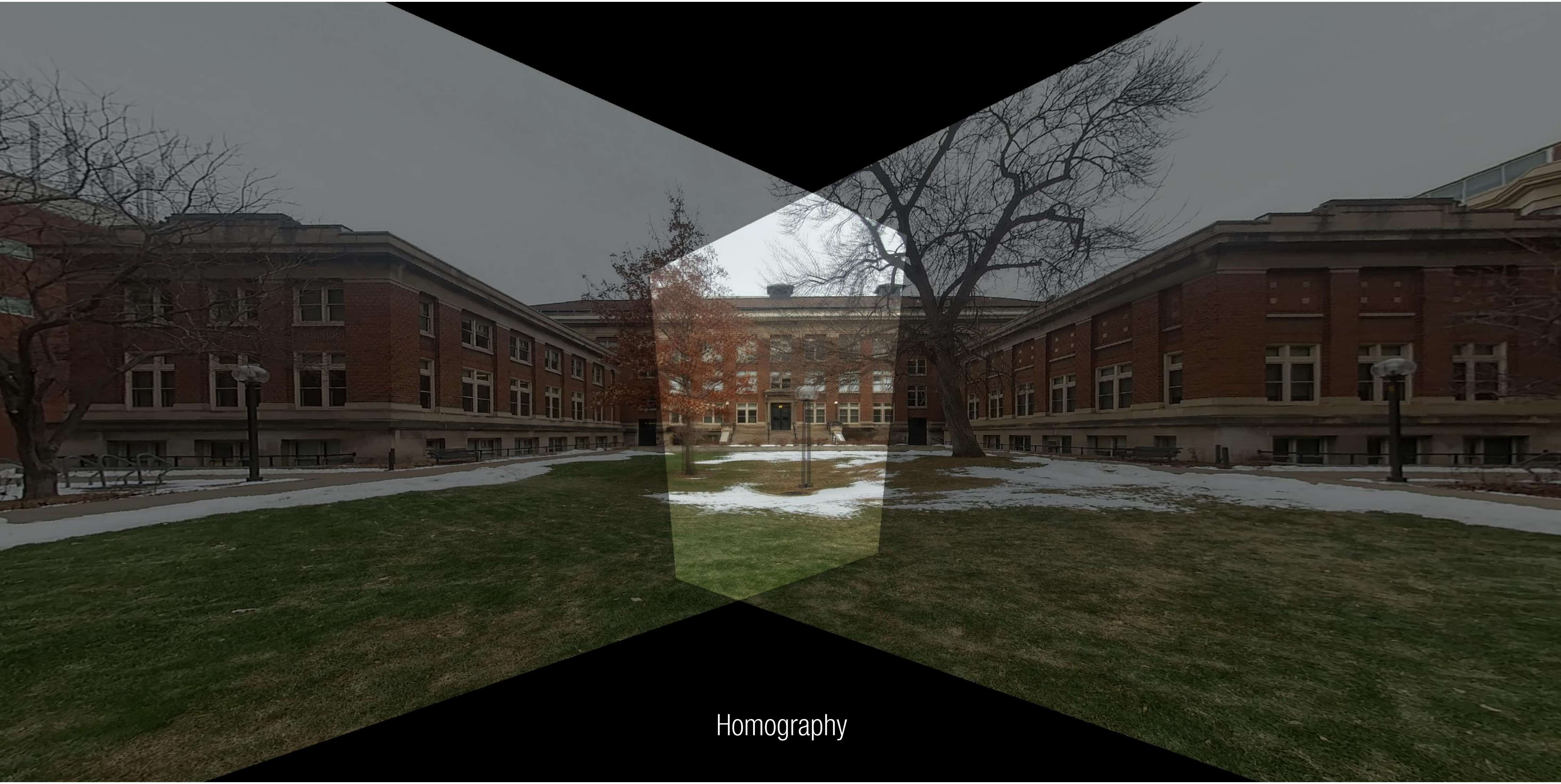
Rotation from homography

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

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)
```

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

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

detR =
0.2059

LindHallCompositeHomography.m

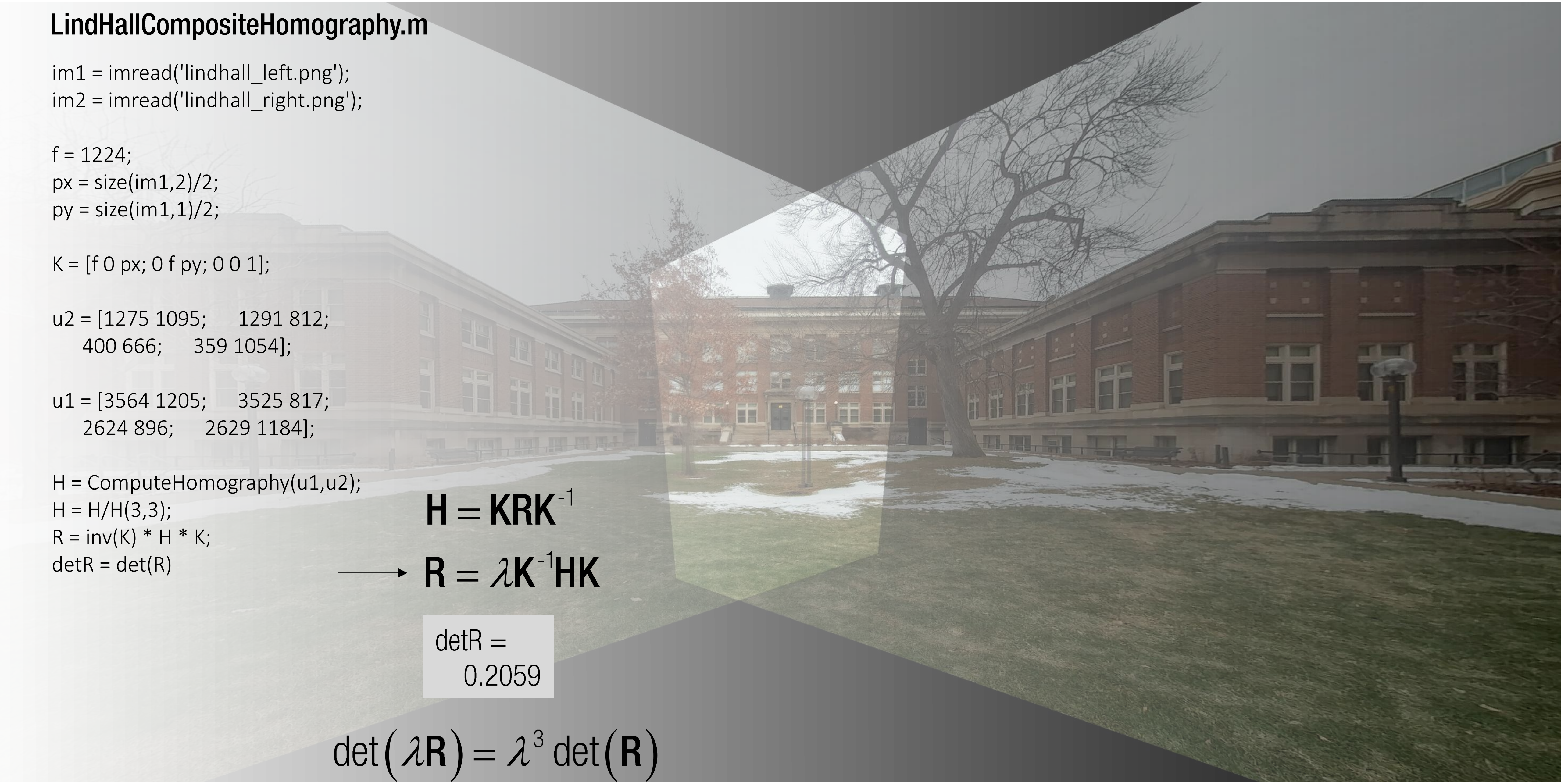
```
im1 = imread('lindhall_left.png');
im2 = imread('lindhall_right.png');

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)
```

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

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

detR =
0.2059

$$\det(\lambda \mathbf{R}) = \lambda^3 \det(\mathbf{R})$$



LindHallCompositeHomography.m

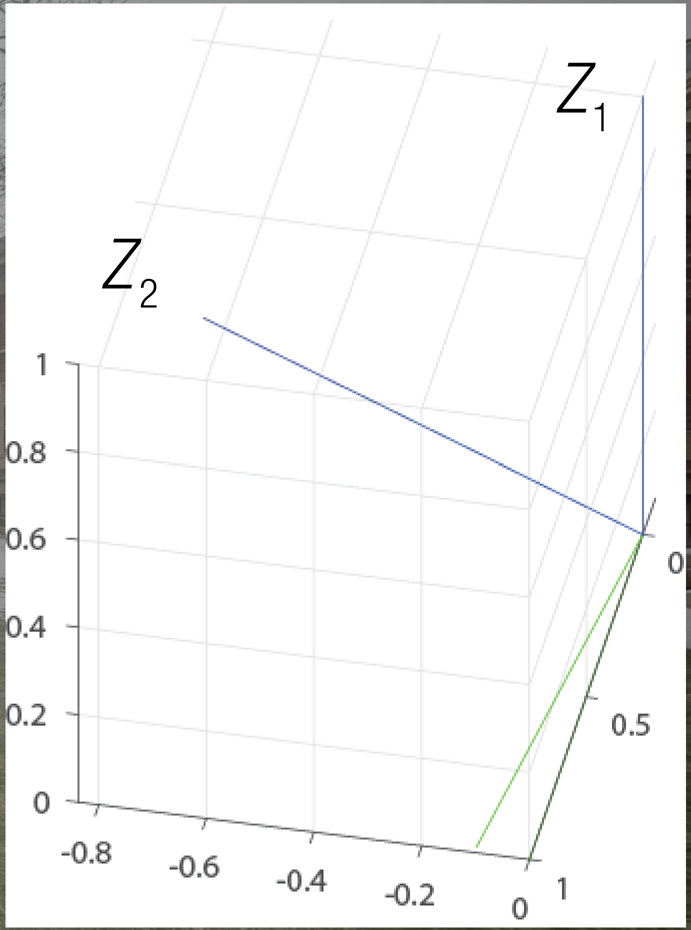
```
im1 = imread('lindhall_left.png');
im2 = imread('lindhall_right.png');

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)
```

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

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

detR =
1.000

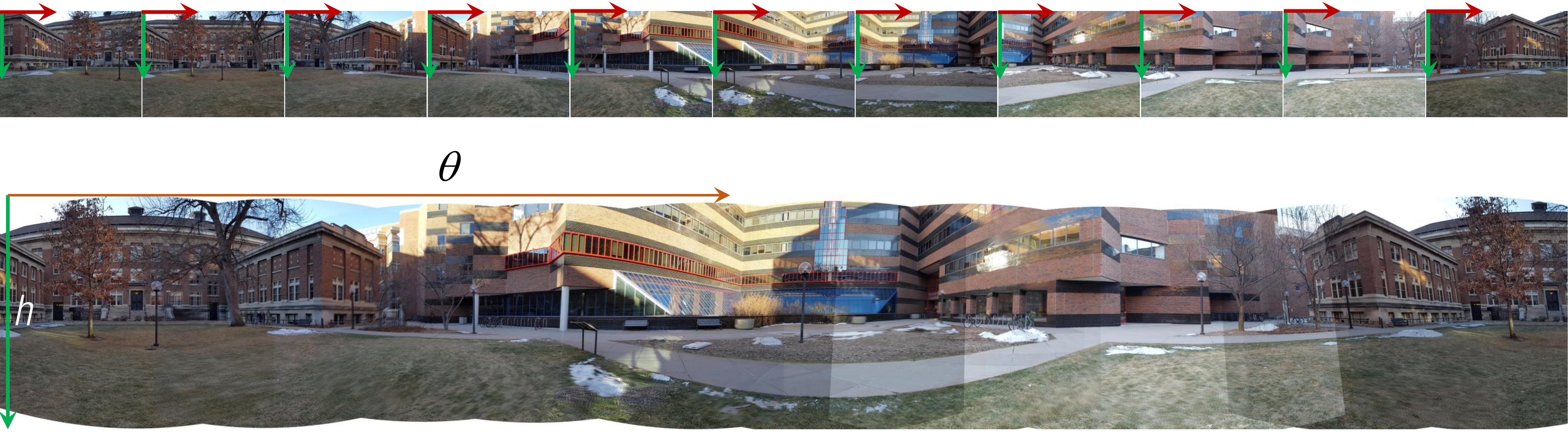
$$\det(\lambda \mathbf{R}) = \lambda^3 \det(\mathbf{R})$$



HW #2: Image Panorama (Cylindrical Projection)

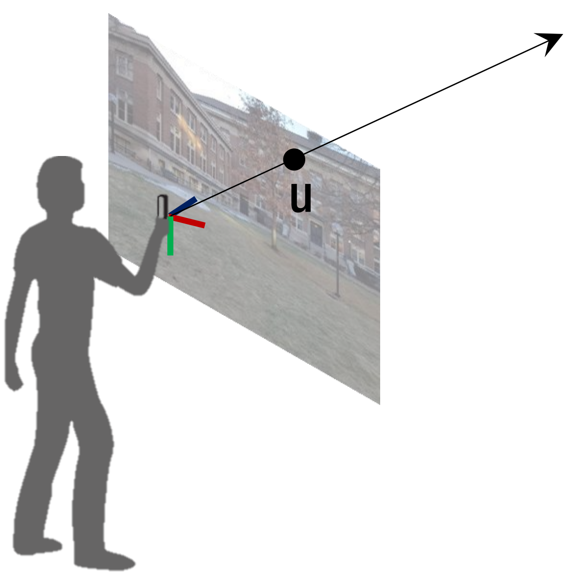
HW #2: Image Panorama (Cylindrical Projection)

HW #2: Image Panorama (Cylindrical Projection)

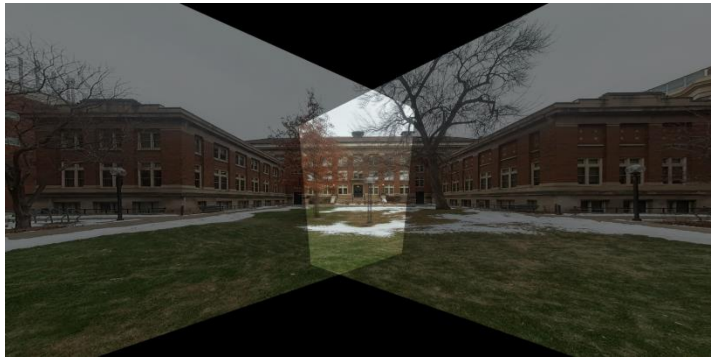


HW #2: Image Panorama (Cylindrical Projection)

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9 R_{10} R_{11}

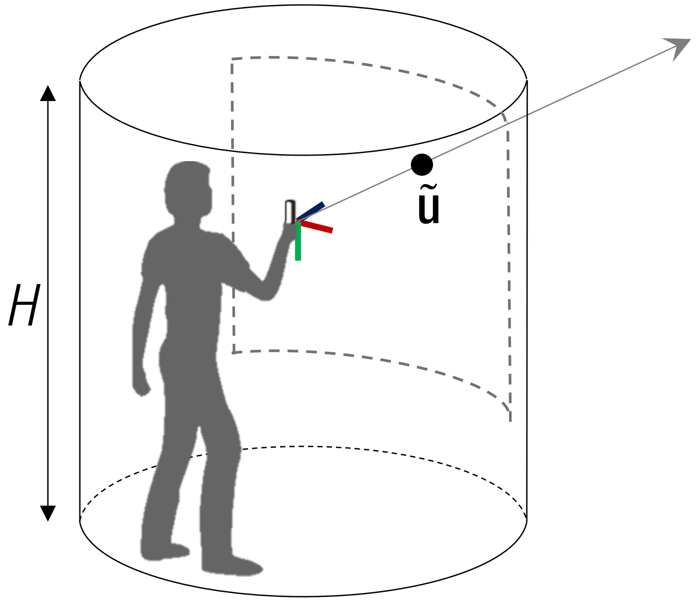


$$\mathbf{X} = \lambda \mathbf{K}^{-1}\mathbf{R}^{\top}\mathbf{u}$$

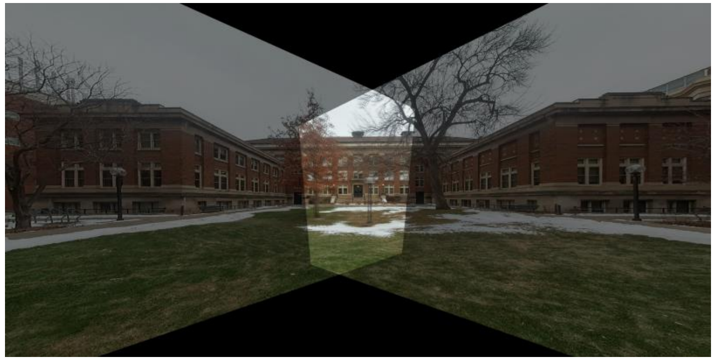


$$\mathbf{R} = \mathbf{K}^{-1}\mathbf{H}\mathbf{K}$$

HW #2: Image Panorama (Cylindrical Projection)



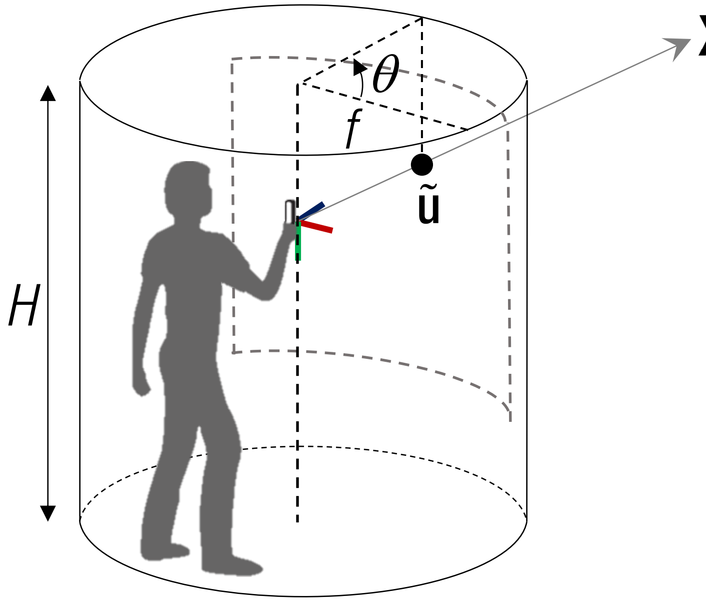
$\mathbf{X} = \lambda \mathbf{K}^{-1}\mathbf{R}^{\top}\mathbf{u}$



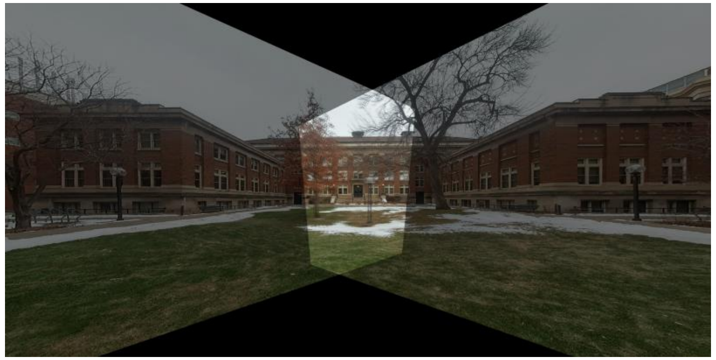
$\mathbf{R} = \mathbf{K}^{-1}\mathbf{H}\mathbf{K}$

$\tilde{\mathbf{u}} =$

HW #2: Image Panorama (Cylindrical Projection)



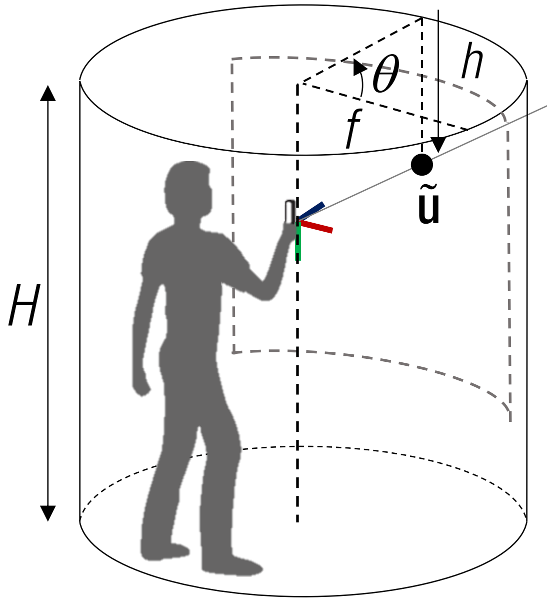
$$\mathbf{X} = \lambda \mathbf{K}^{-1}\mathbf{R}^{\top}\mathbf{u}$$



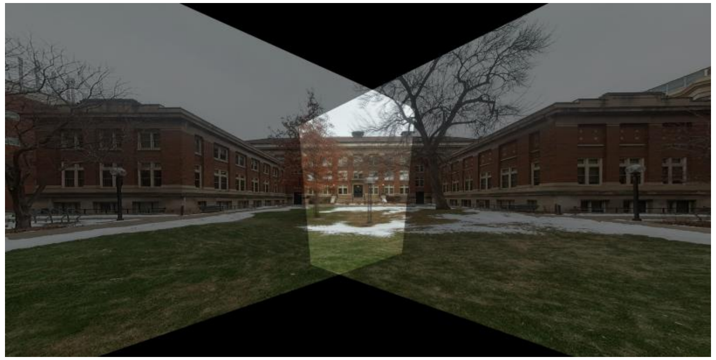
$$\mathbf{R} = \mathbf{K}^{-1}\mathbf{H}\mathbf{K}$$

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

HW #2: Image Panorama (Cylindrical Projection)



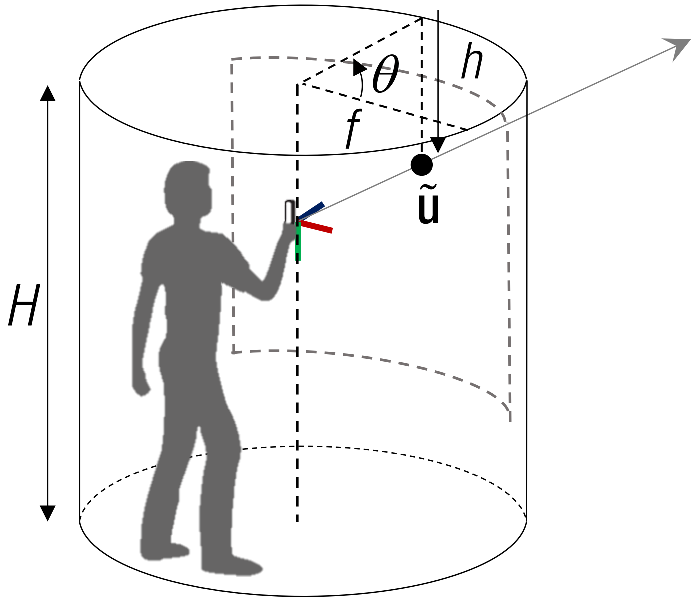
$$\mathbf{X} = \lambda \mathbf{K}^{-1}\mathbf{R}^{\top}\mathbf{u}$$



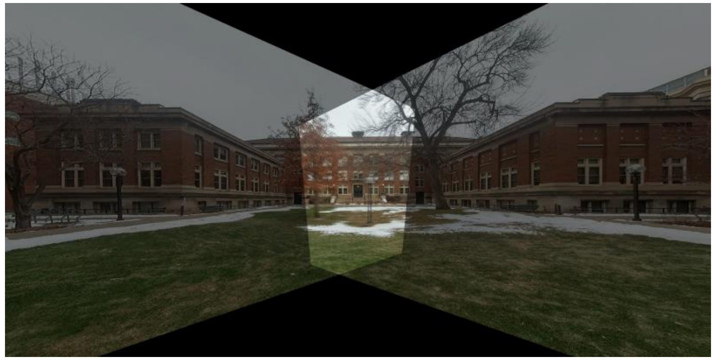
$$\mathbf{R} = \mathbf{K}^{-1}\mathbf{H}\mathbf{K}$$

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

HW #2: Image Panorama (Cylindrical Projection)



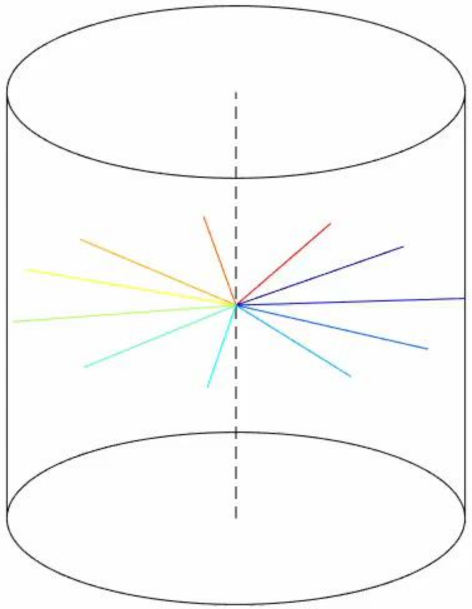
$$\mathbf{X} = \lambda \mathbf{K}^{-1}\mathbf{R}^{\mathsf{T}}\mathbf{u}$$



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

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

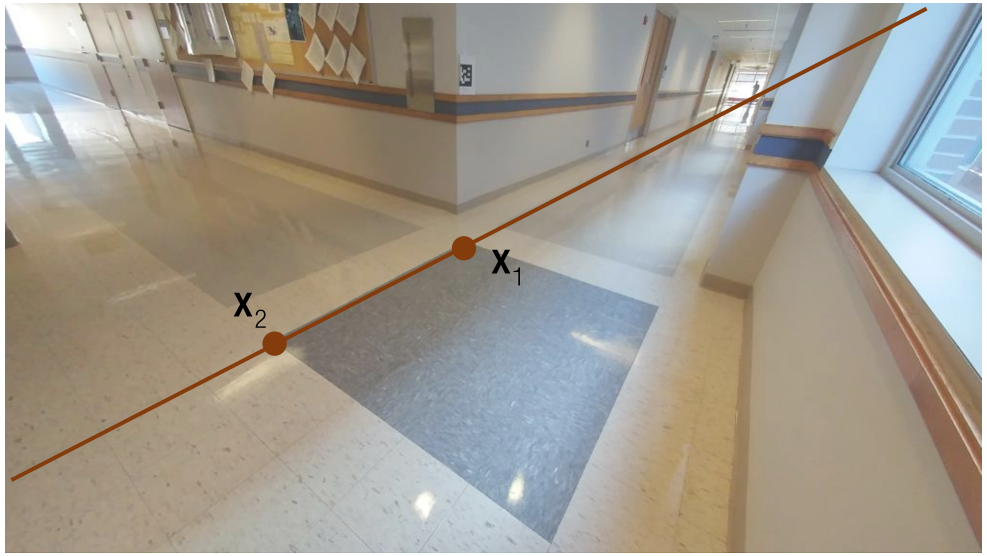
HW #2: Image Panorama (Cylindrical Projection)



How to compute homography?

Linear Parameter Estimation

Point-Point in Image

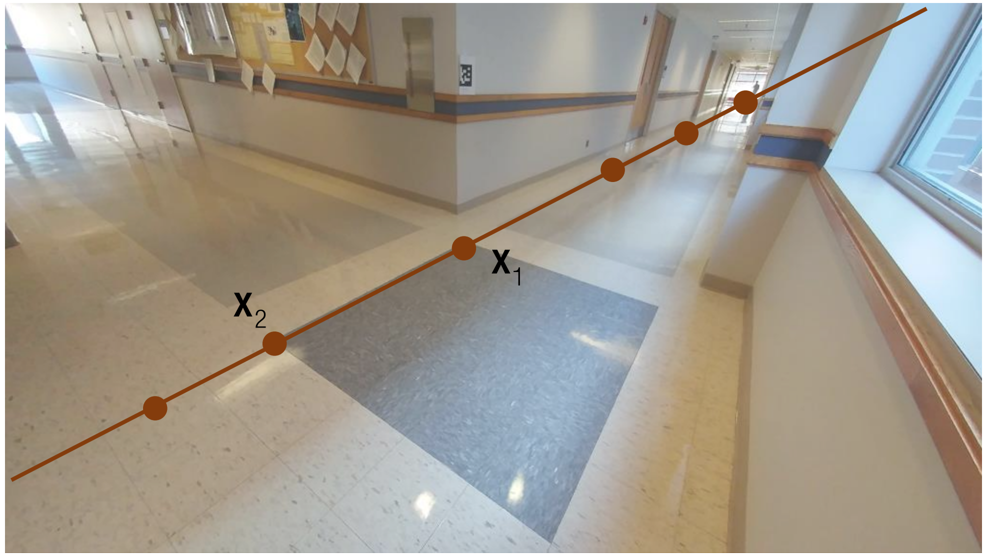


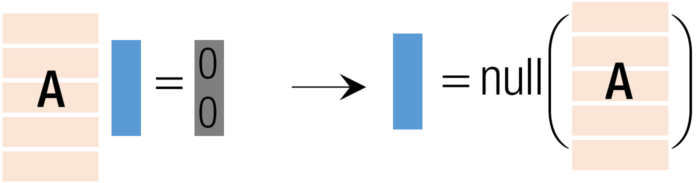
$$\begin{bmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \end{bmatrix} \mathbf{l} = \mathbf{0}$$

$$\mathbf{A} \, \blacksquare = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \longrightarrow \blacksquare = \mathrm{null}\left(\mathbf{A}\right) \quad \text{or} \quad \mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$$

3x2

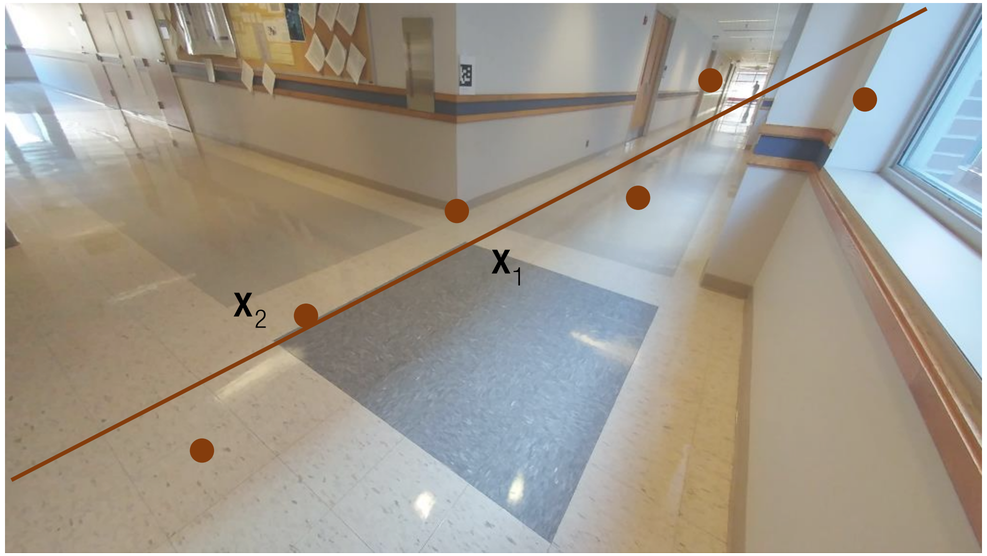
Point-Point in Image

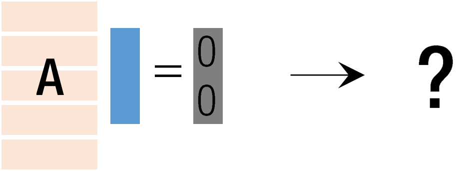




or $\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$

Line Fitting

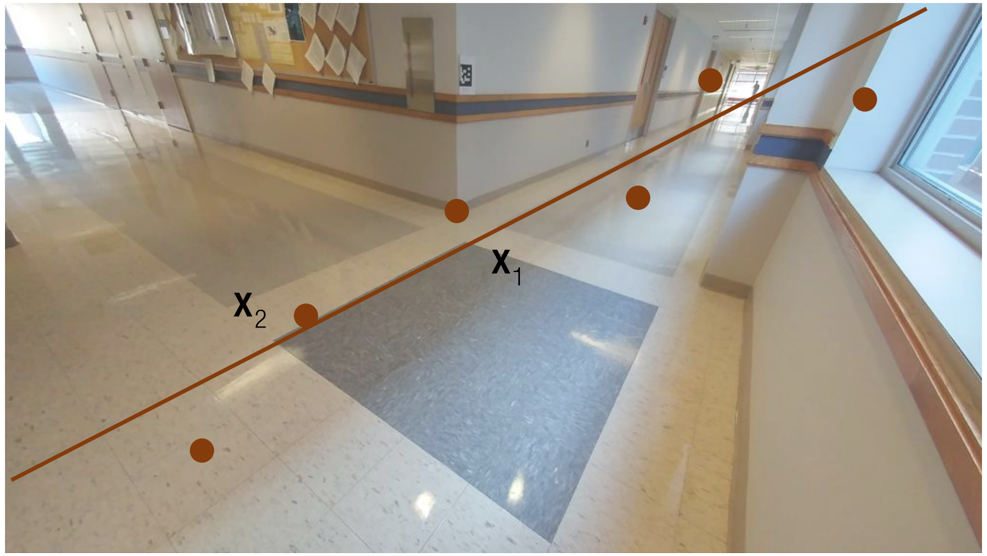




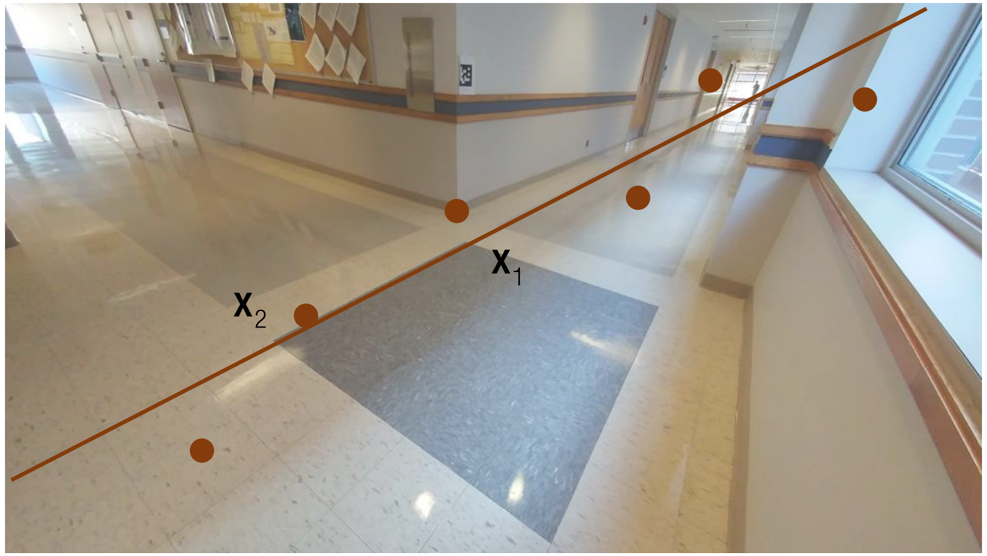
Line Fitting

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

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



Line Fitting



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

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

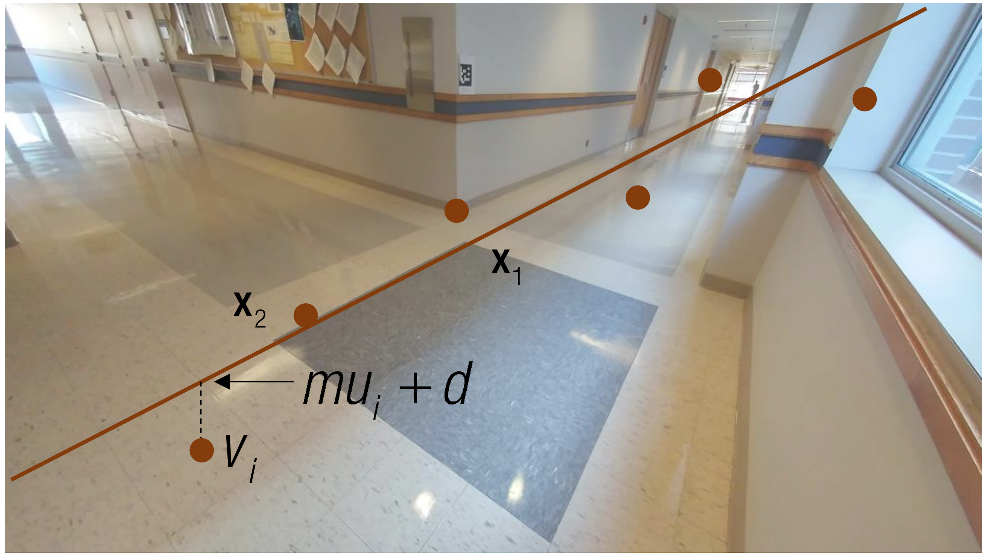
$$\longrightarrow au_1 + bv_1 + c \approx 0$$

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

$$\vdots$$

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

Line Fitting



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

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

slope y-intercept

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

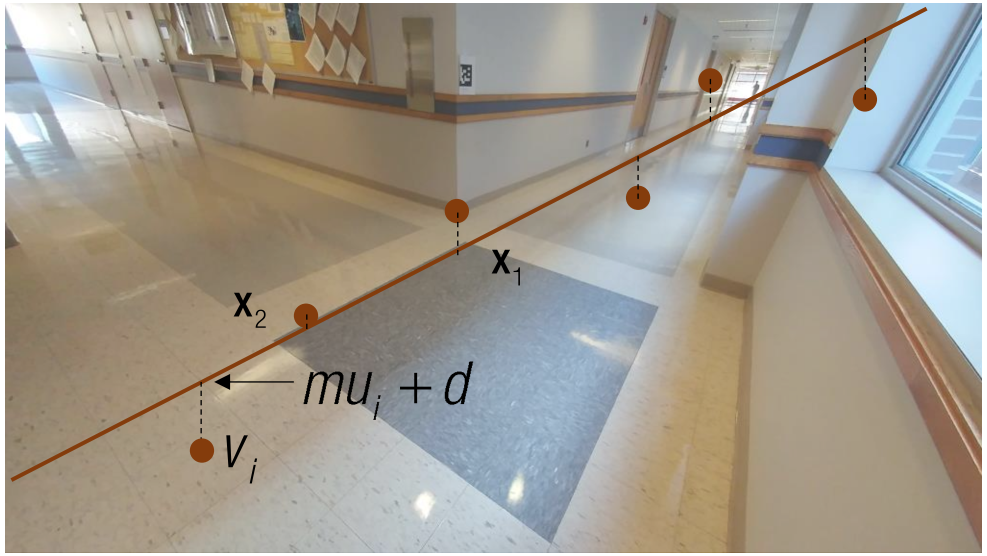
Line Fitting

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

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

slope y-intercept

$$\longrightarrow \quad v_1 \approx mu_1 + d$$
$$v_2 \approx mu_2 + d$$
$$\vdots$$
$$v_n \approx mu_n + d$$



Error: $e_i = v_i - (mu_i + d)$

$$e_1 = v_1 - (mu_1 + d)$$
$$e_2 = v_2 - (mu_2 + d)$$

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

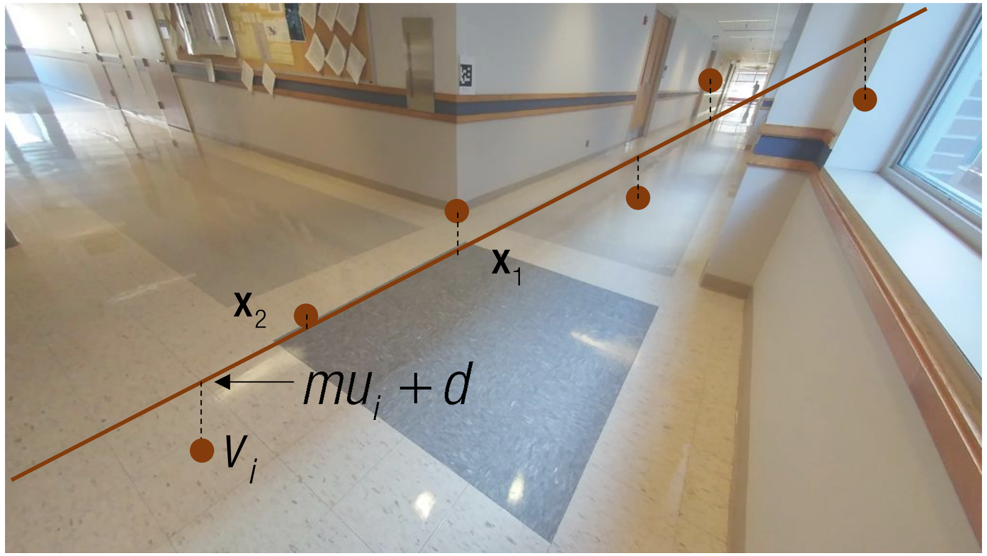
Line Fitting

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

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

slope y-intercept

$$\longrightarrow \quad v_1 \approx mu_1 + d$$
$$v_2 \approx mu_2 + d$$
$$\vdots$$
$$v_n \approx mu_n + d$$



$$e_1 = v_1 - (mu_1 + d)$$
$$e_2 = v_2 - (mu_2 + d)$$

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

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

Unknowns:

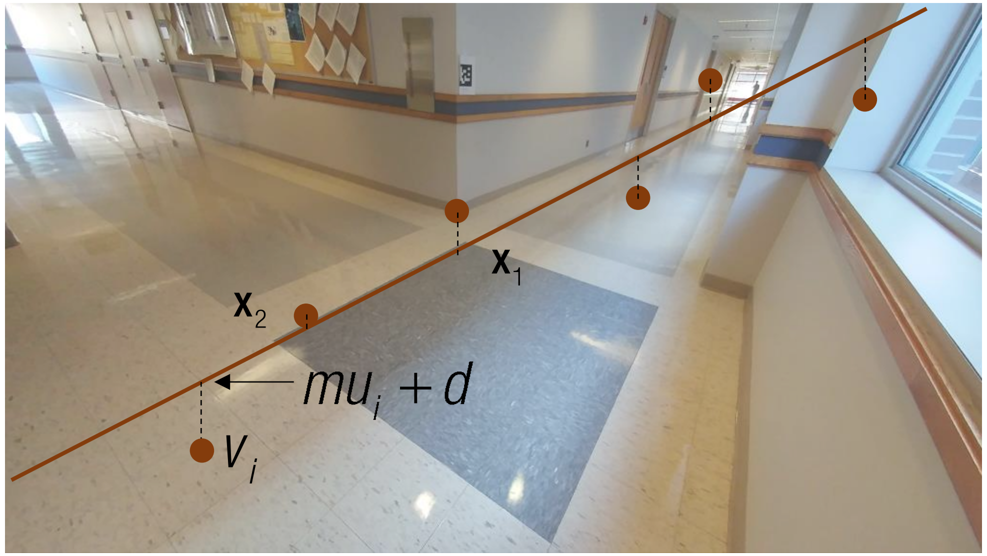
Number of eq.:

Line Fitting

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

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

slope   y-intercept



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

$$v_2 \approx mu_2 + d$$

$$\vdots$$

$$v_n \approx mu_n + d$$

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

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

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

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

Unknowns: m,d

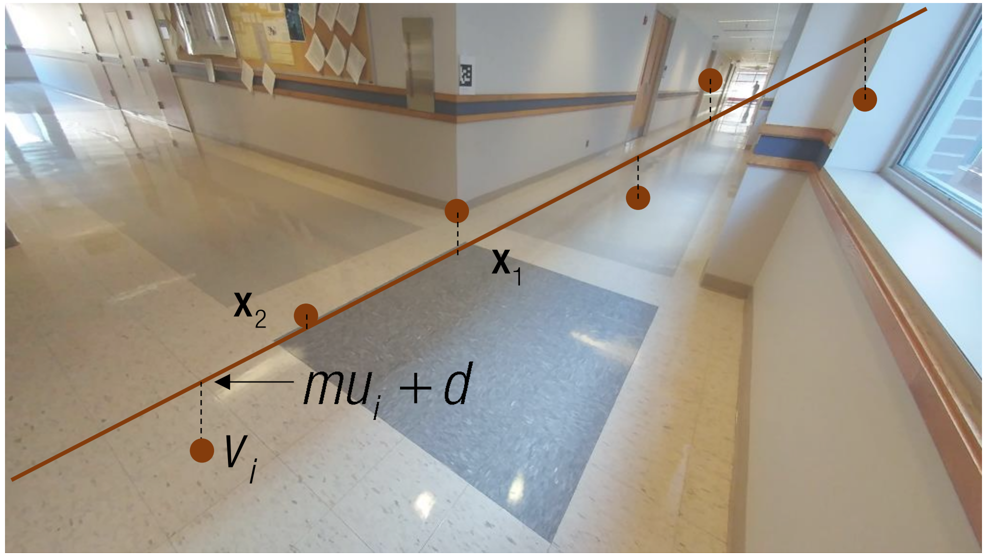
Number of eq.:

Line Fitting

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

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

slope y-intercept



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

$$v_2 \approx mu_2 + d$$

$$\vdots$$

$$v_n \approx mu_n + d$$

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

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

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

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

Unknowns: m,d

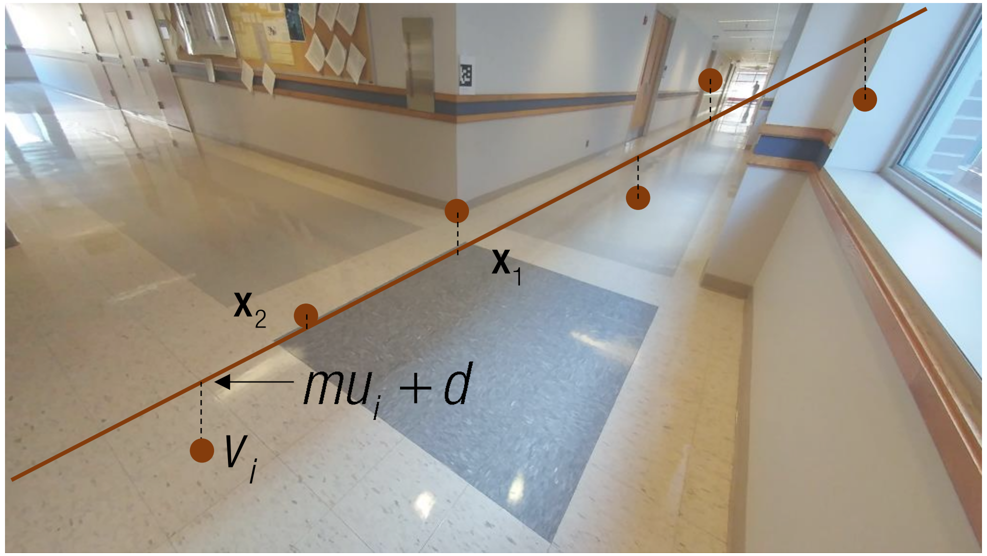
Number of eq.: n

Line Fitting

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

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

slope y-intercept



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

$$v_2 \approx mu_2 + d$$

$$\vdots$$

$$v_n \approx mu_n + d$$

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

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

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

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

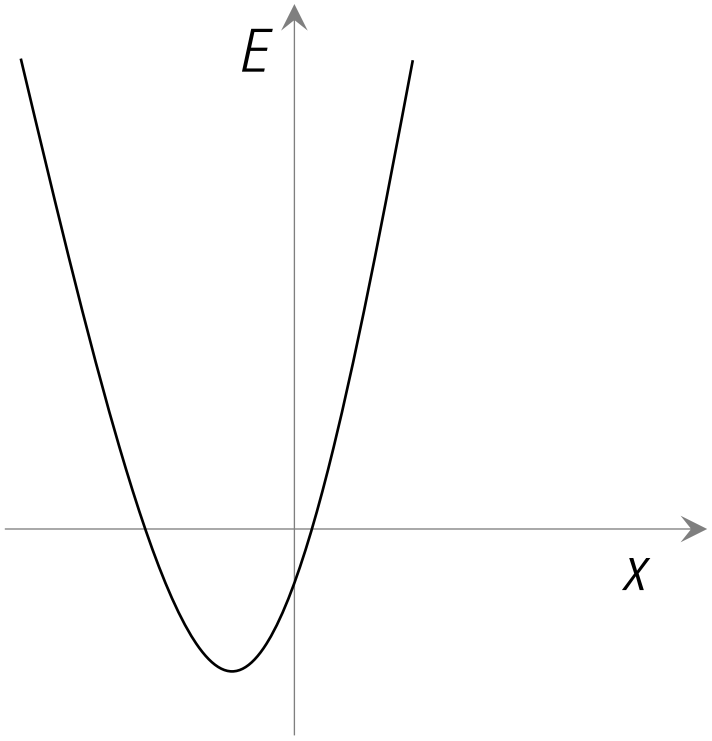
Unknowns: m,d

Number of eq.: n

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

How to minimize?

Line Fitting



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

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

slope y-intercept

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

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

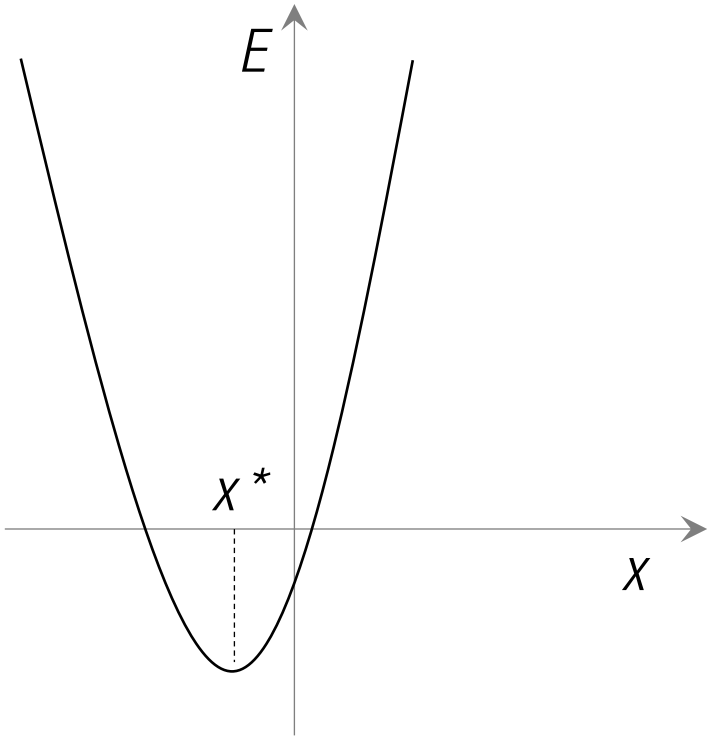
Unknowns: m, d

Number of eq.: n

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

How to minimize?

Line Fitting



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

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

slope y-intercept

$$\longrightarrow \quad v_1 \approx mu_1 + d$$
$$v_2 \approx mu_2 + d$$
$$\vdots$$
$$v_n \approx mu_n + d$$

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

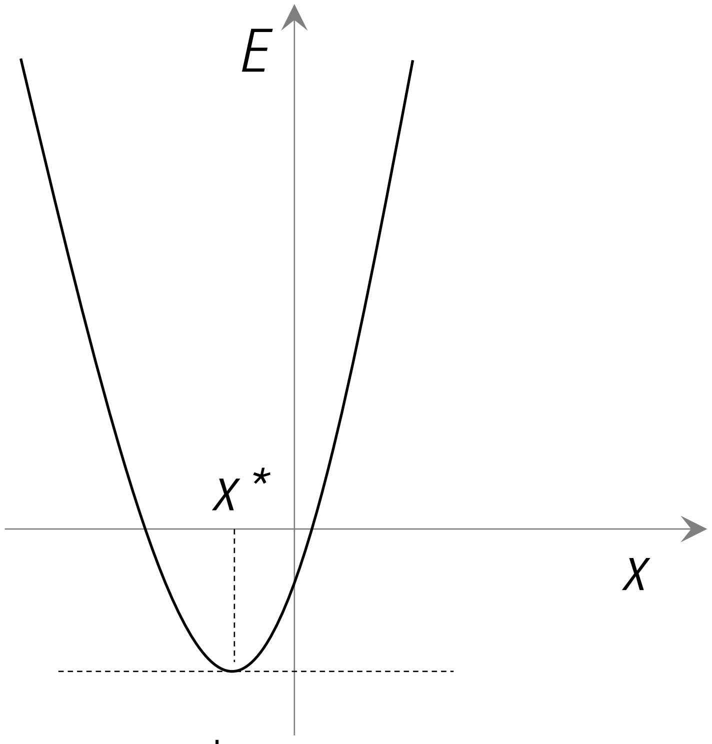
Unknowns: m,d

Number of eq.: n

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

How to minimize?

Line Fitting



$$\left.\frac{dE}{dx}\right|_{x=x^*}=0$$

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

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

slope y-intercept

$$\longrightarrow \quad v_1 \approx mu_1 + d$$
$$v_2 \approx mu_2 + d$$
$$\vdots$$
$$v_n \approx mu_n + d$$

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

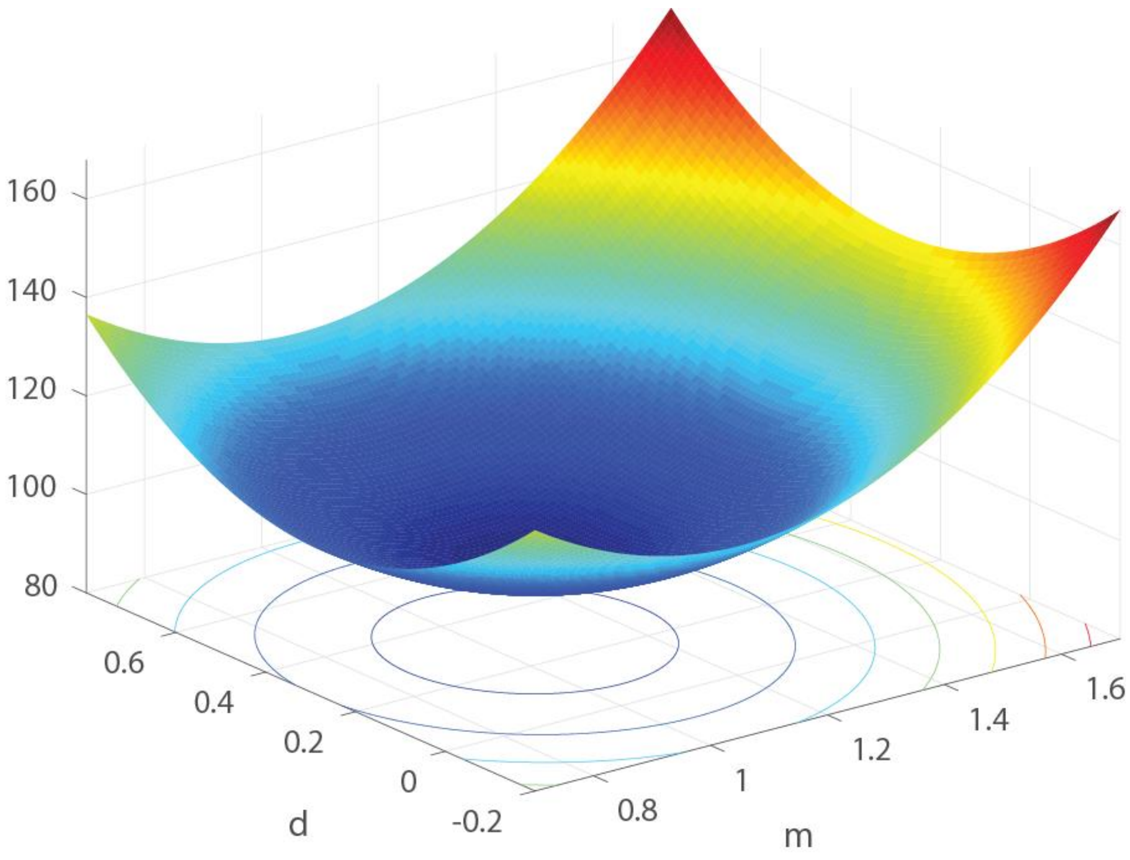
Unknowns: m,d

Number of eq.: n

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

How to minimize?

Line Fitting



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

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

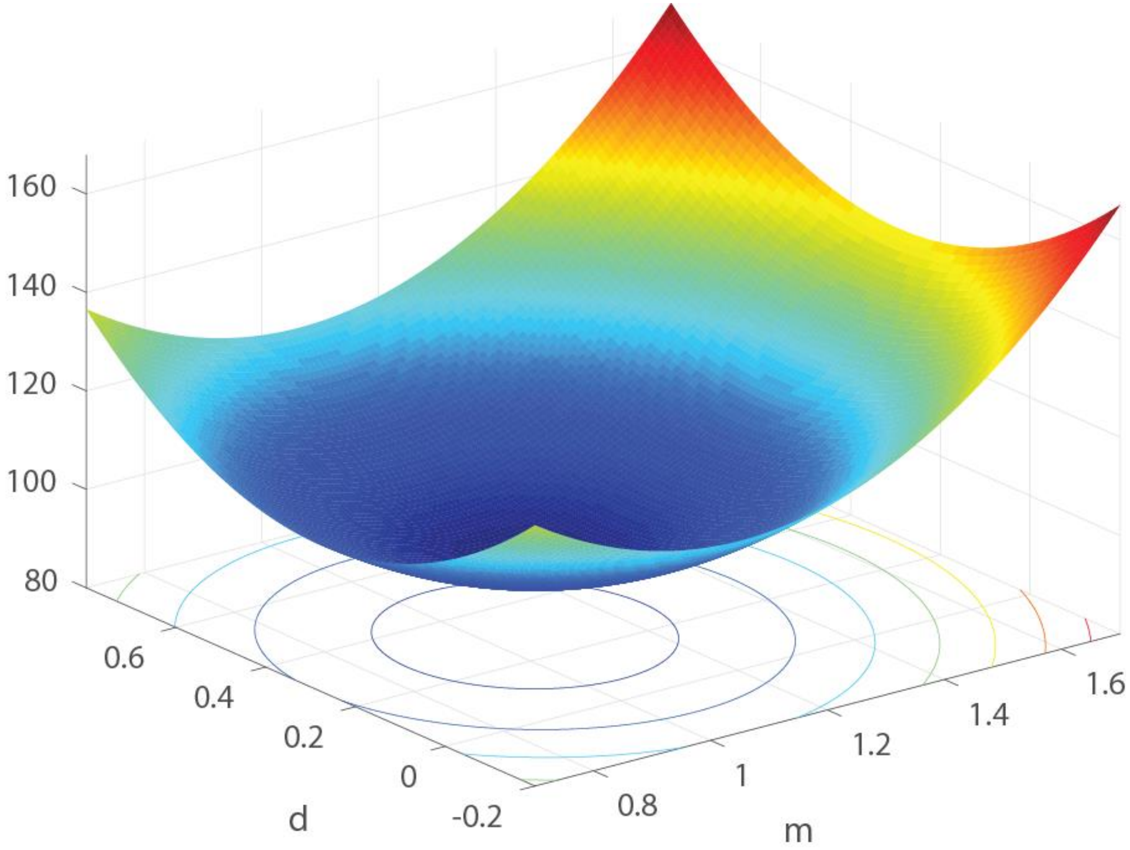
$$\frac{\partial E}{\partial m} =$$

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

Line Fitting

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

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



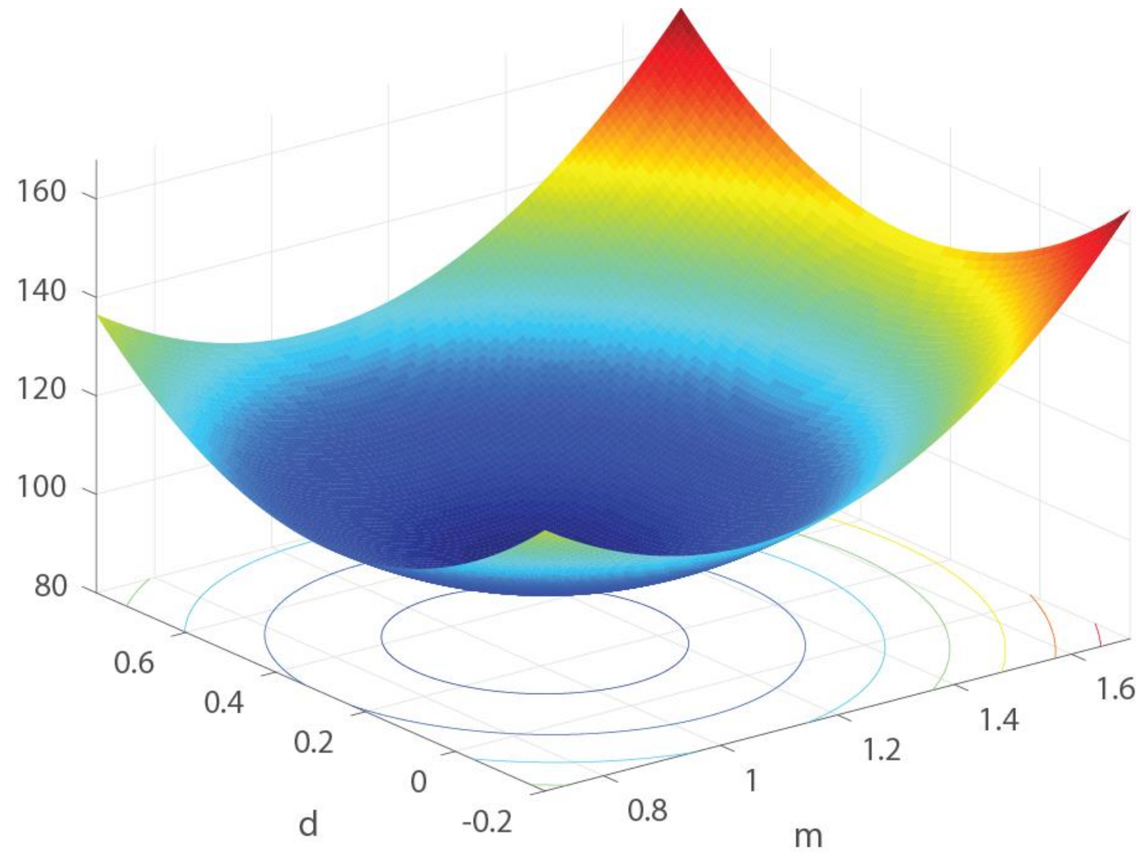
$$\frac{\partial E}{\partial m} = -\sum_{i=1}^{n} 2u_i \left(v_i - (mu_i + d)\right) = 0$$

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

Line Fitting

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

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



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

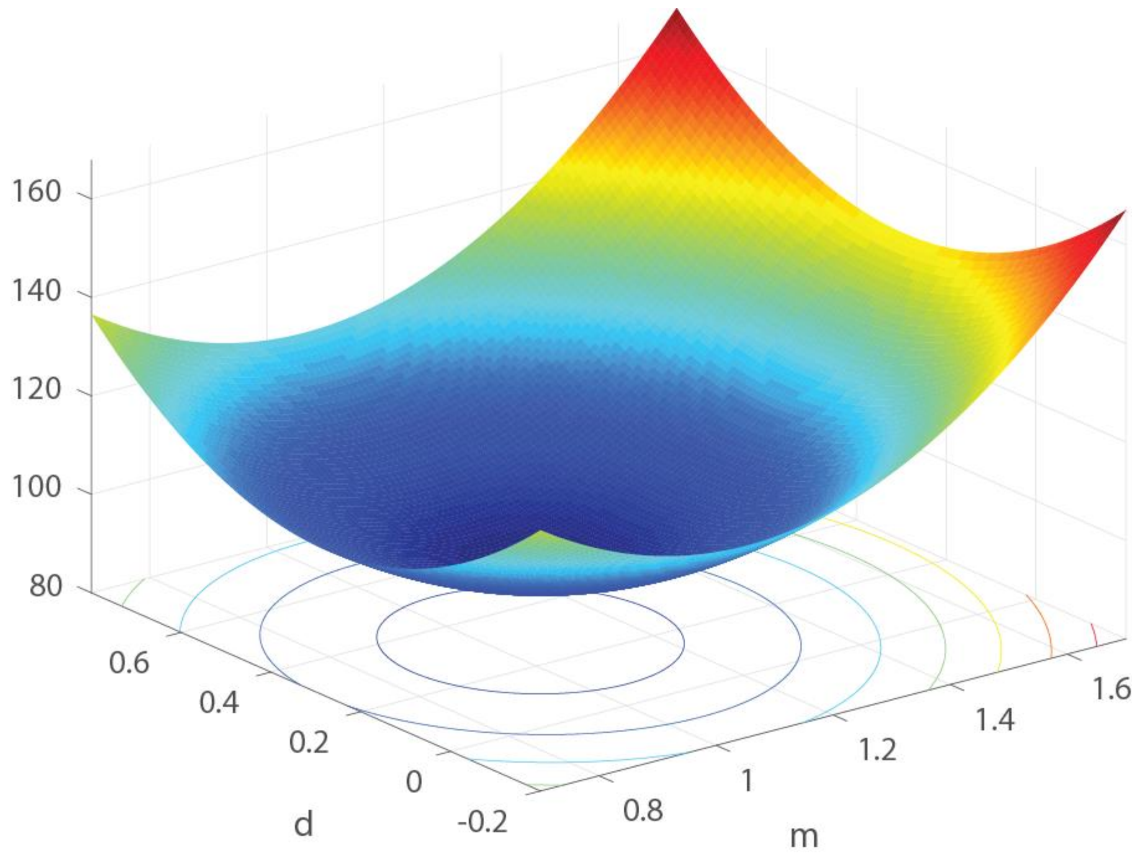
$$\frac{\partial E}{\partial d} = -\sum_{i=1}^{n} 2\left(v_i - (mu_i + d)\right) = 0$$

$\longrightarrow$

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

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

Line Fitting



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

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

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

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

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

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

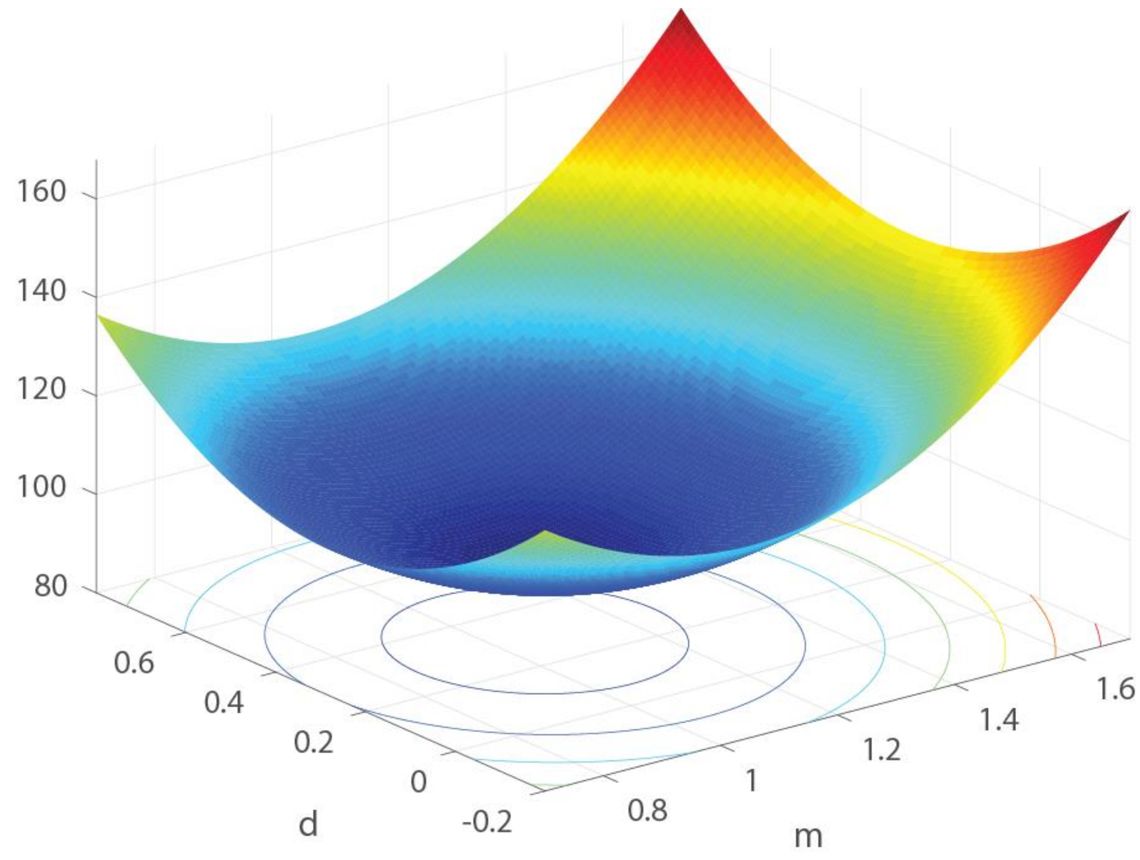
$$\begin{bmatrix} \sum_{i=1}^{n} u_i^2 & \sum_{i=1}^{n} u_i \\ \sum_{i=1}^{n} u_i & n \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} u_i v_i \\ \sum_{i=1}^{n} v_i \end{bmatrix}$$

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

Line Fitting (Ax=b)

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

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

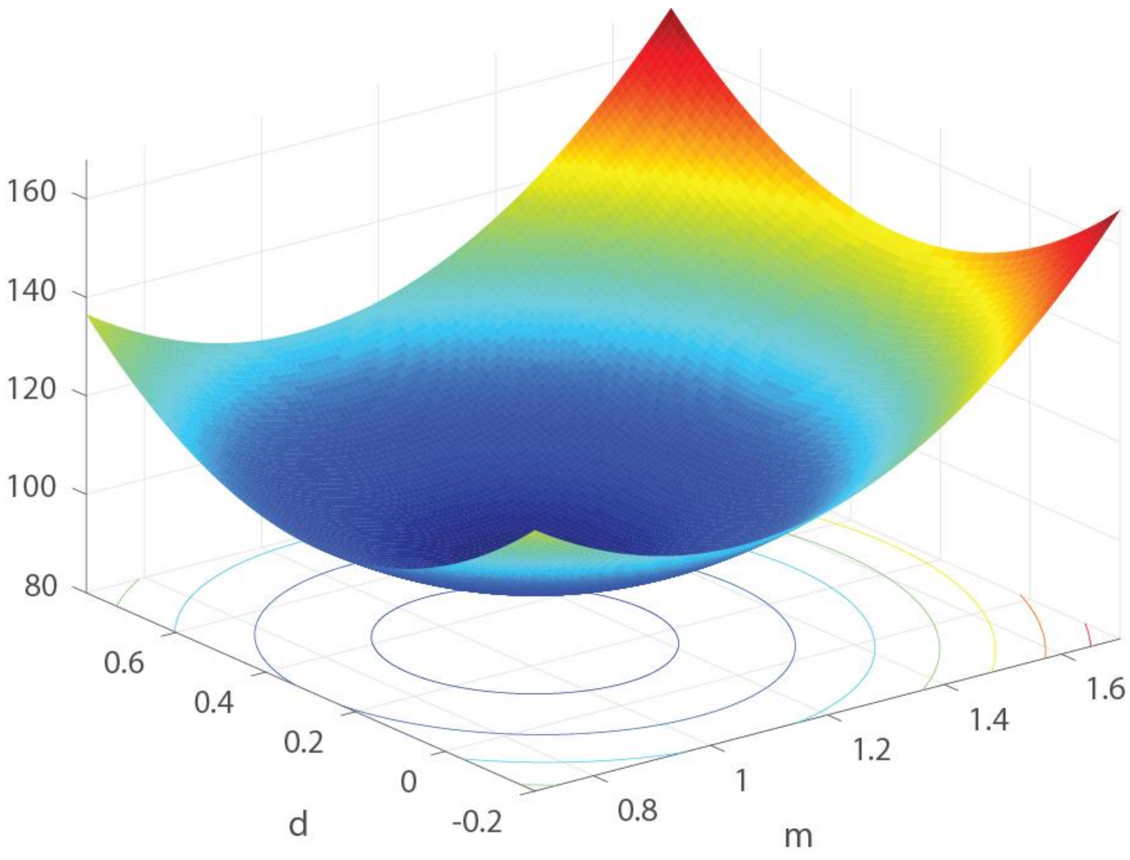


$v_1 \approx mu_1 + d$

$v_2 \approx mu_2 + d$

$v_n \approx mu_n + d$

Line Fitting (Ax=b)



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

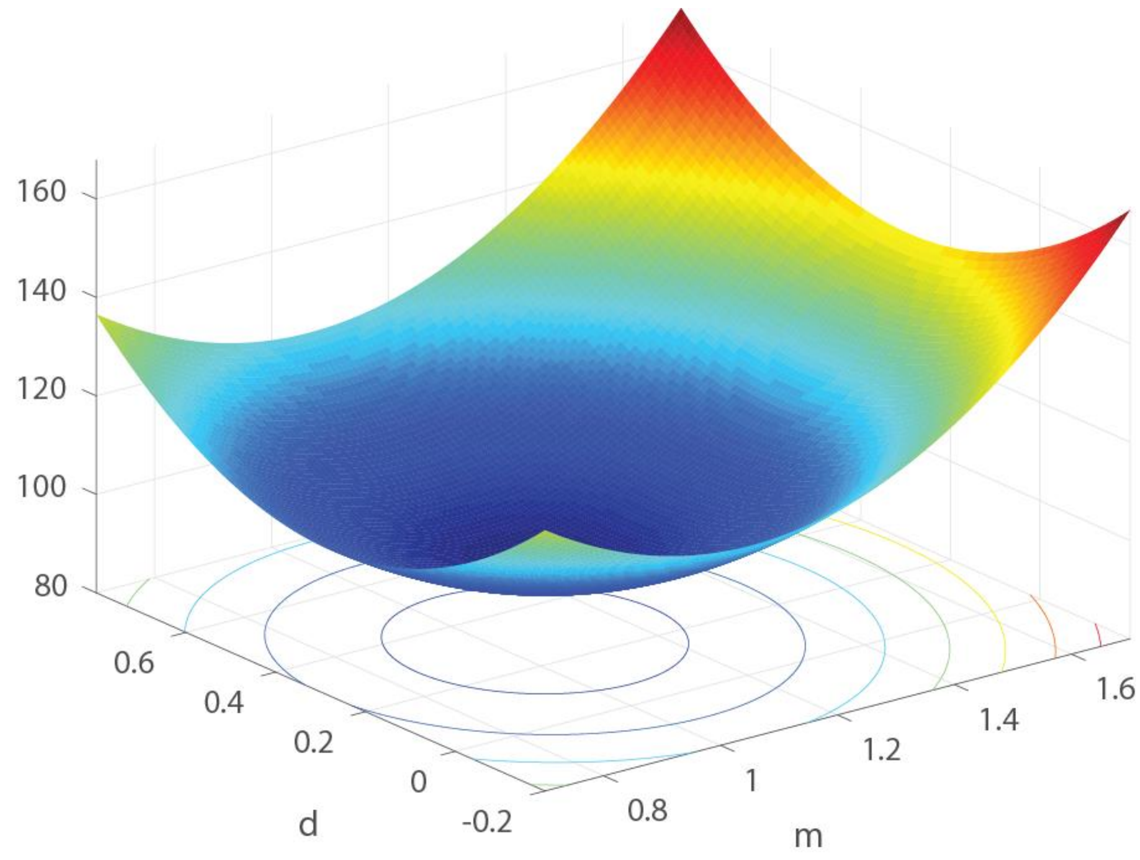
Optimal point: $\frac{\partial E}{\partial m} = 0, \quad \frac{\partial E}{\partial d} = 0$

$$v_1 \approx mu_1 + d \qquad \begin{bmatrix} u_1 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_1$$

$$v_2 \approx mu_2 + d \quad \longrightarrow \quad \begin{bmatrix} u_2 & 1 \end{bmatrix} \begin{bmatrix} m \\ d \end{bmatrix} \approx v_2$$

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

Line Fitting (Ax=b)



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

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

$$v_1 \approx mu_1 + d$$
$$v_2 \approx mu_2 + d$$
$$v_n \approx mu_n + d$$

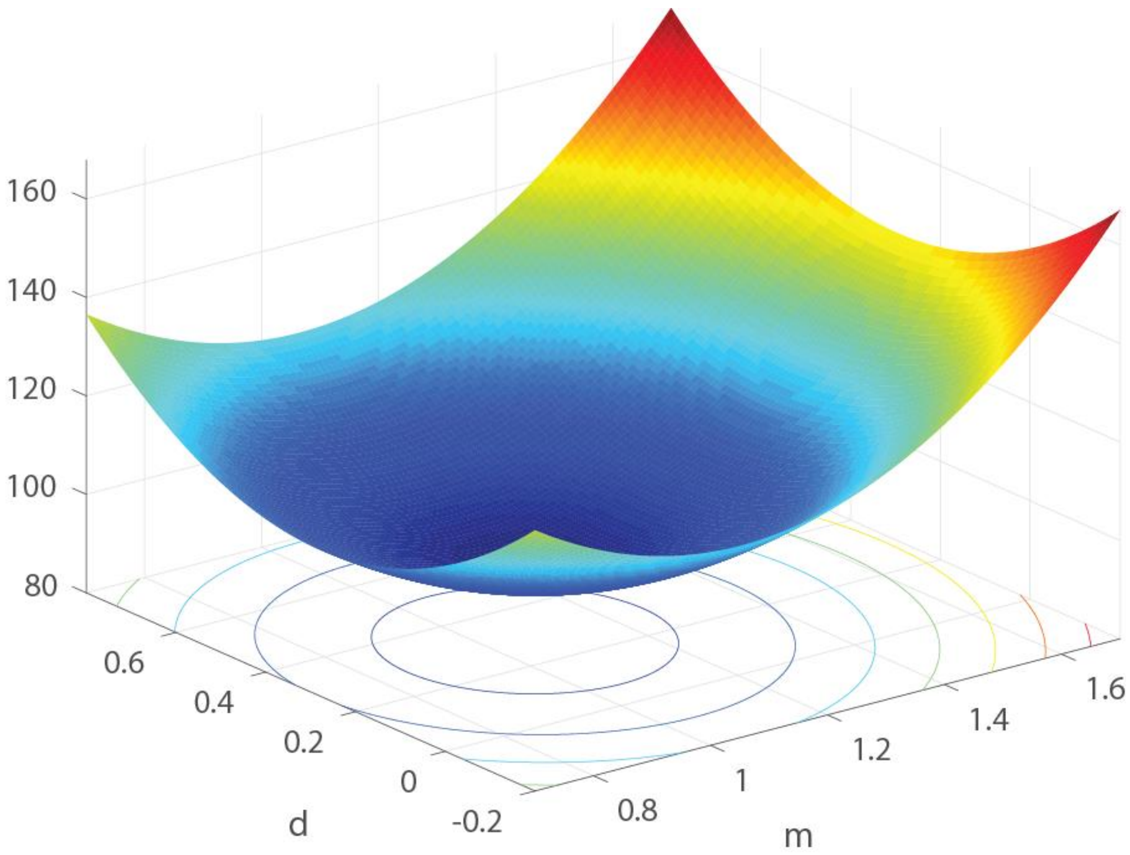
$\rightarrow$

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

$\rightarrow$

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

Line Fitting (Ax=b)



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

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

$$v_1 \approx mu_1 + d$$
$$v_2 \approx mu_2 + d$$
$$v_n \approx mu_n + d$$

$\longrightarrow$

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

$\longrightarrow$

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

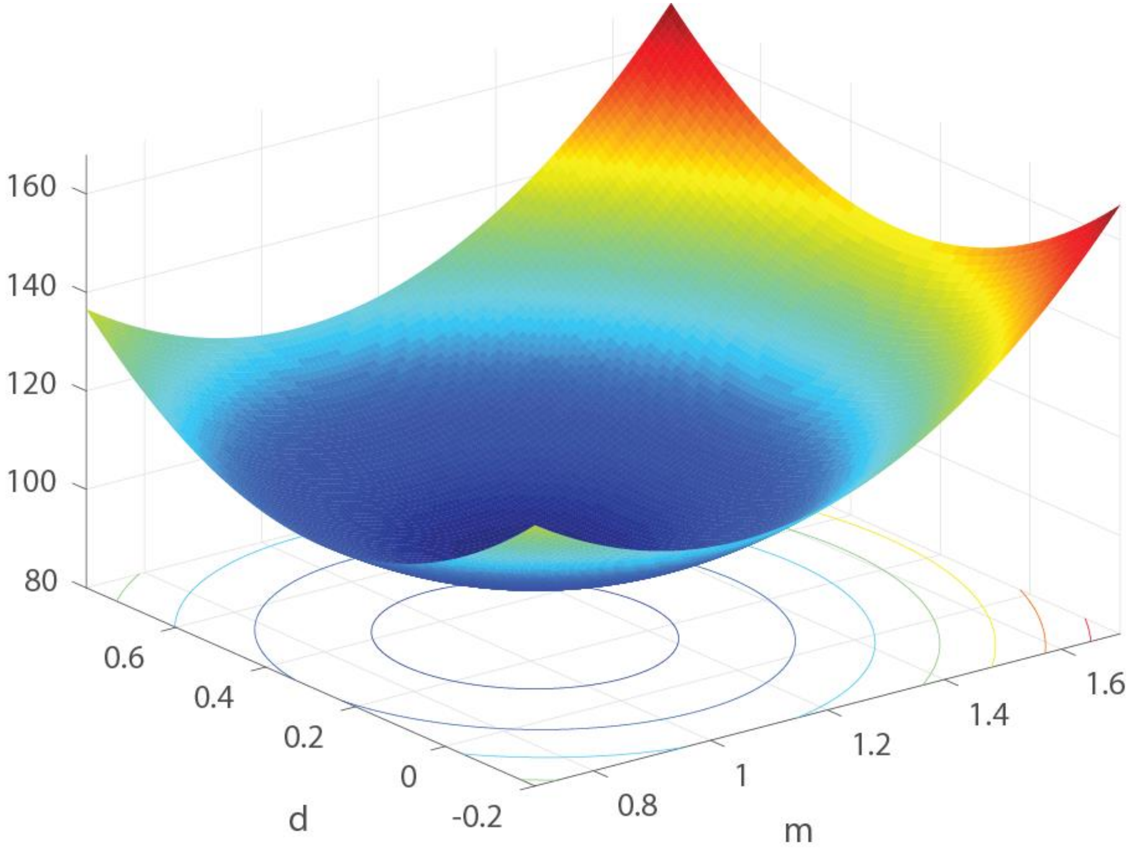
We can't invert **A**.

Line Fitting (Ax=b)

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

$$E = (\mathbf{Ax} - \mathbf{b})^{\mathrm{T}} (\mathbf{Ax} - \mathbf{b}) = \left\| \mathbf{Ax} - \mathbf{b} \right\|^2$$

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

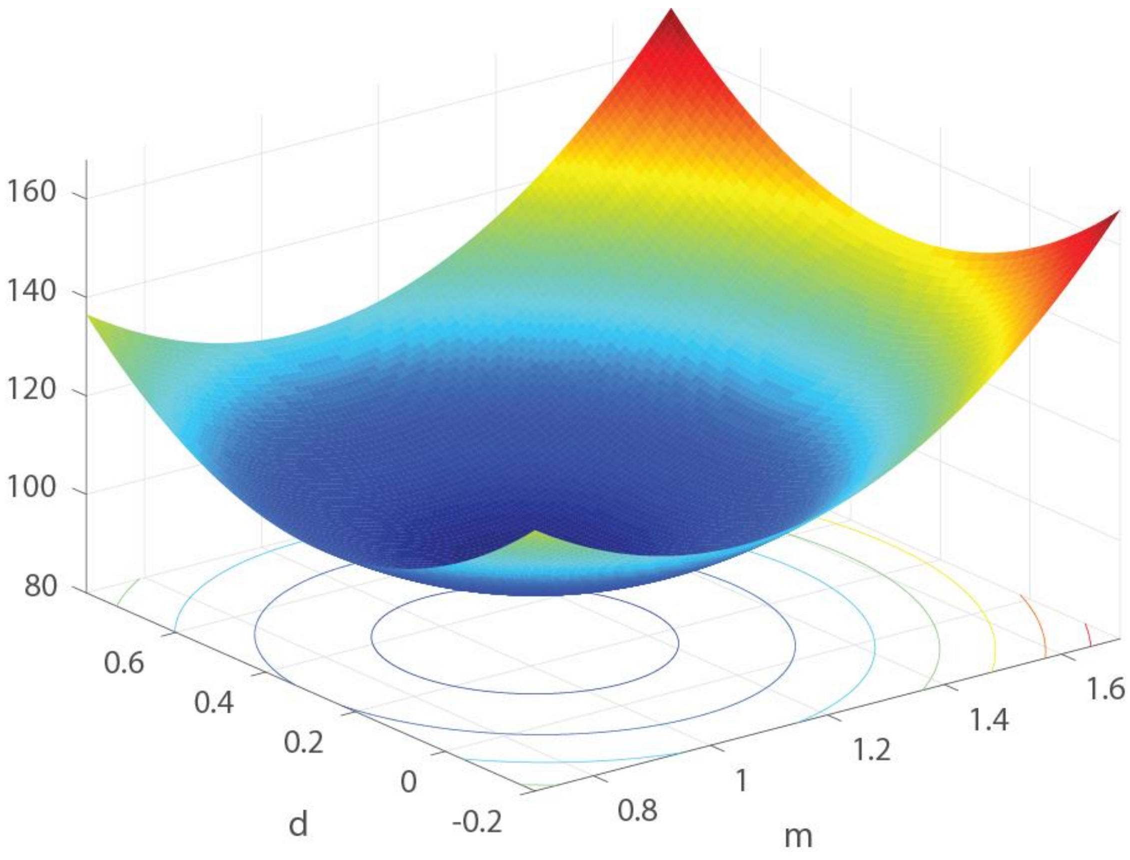


Line Fitting (Ax=b)

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

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

$$E = (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) = \|\mathbf{Ax} - \mathbf{b}\|^2$$
$$= \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{x}^T\mathbf{A}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}$$



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

Line Fitting (Ax=b)

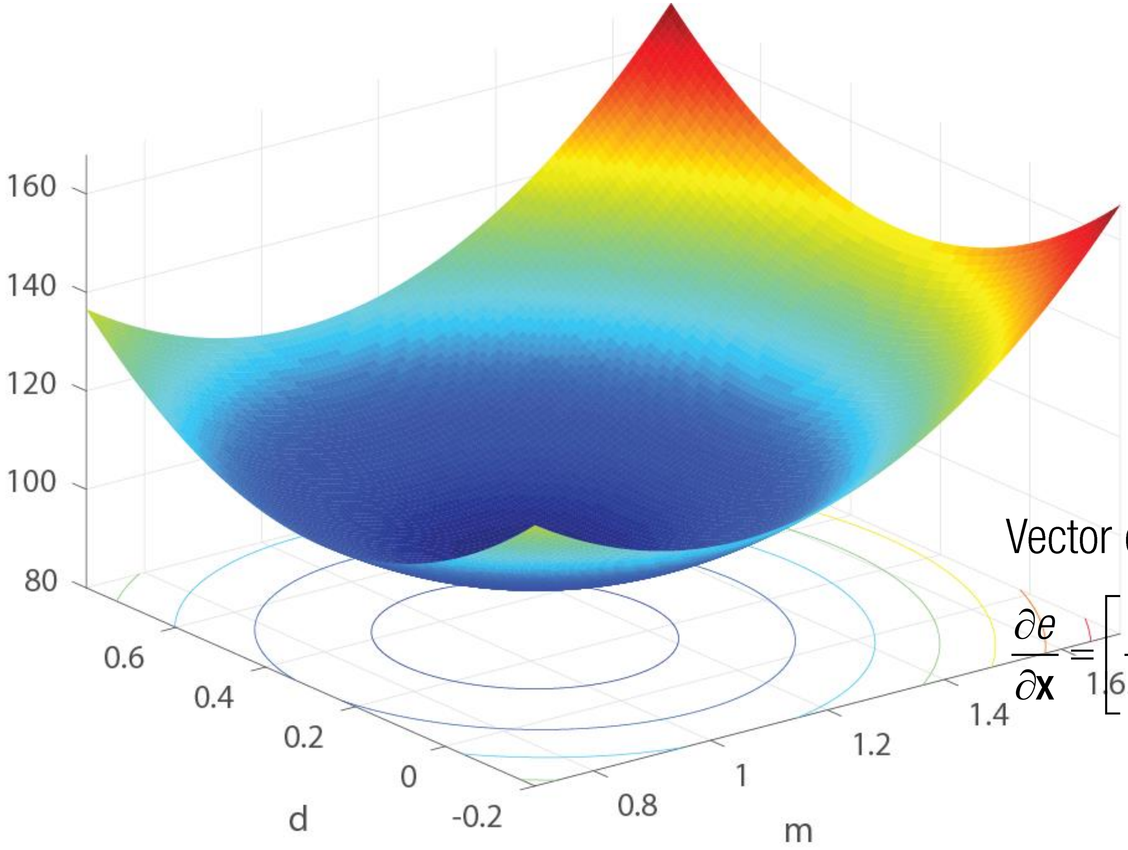
Total error: $E = \sum_{i=1}^{n} \left(v_i - (mu_i + d)\right)^2$

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

$$E = (\mathbf{Ax} - \mathbf{b})^{\mathsf{T}} (\mathbf{Ax} - \mathbf{b}) = \left\|\mathbf{Ax} - \mathbf{b}\right\|^2$$

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

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

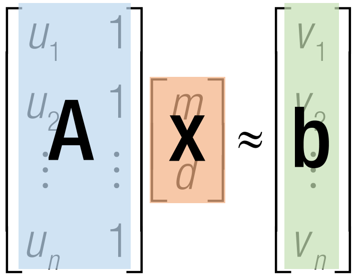


Vector derivative:

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

Line Fitting (Ax=b)

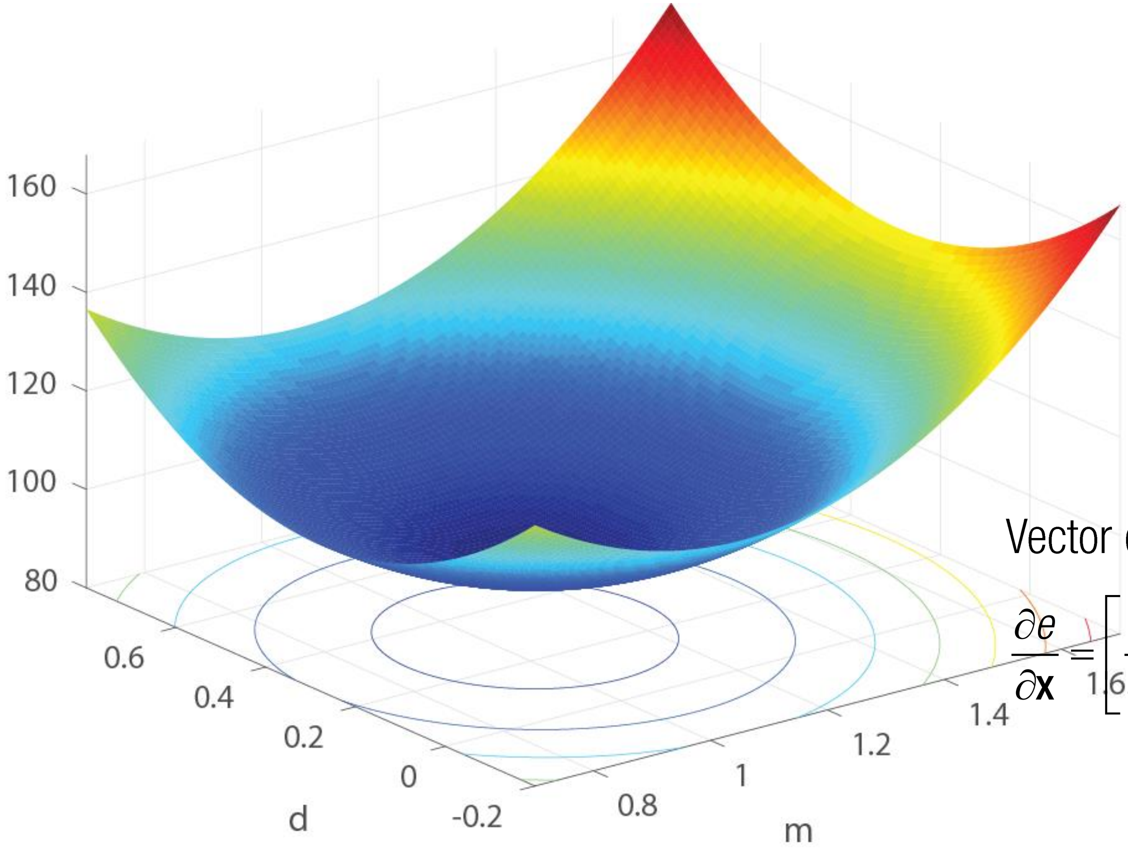
Total error: $E = \sum_{i=1}^{n} \left(v_i - (mu_i + d) \right)^2$



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

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

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



Vector derivative:

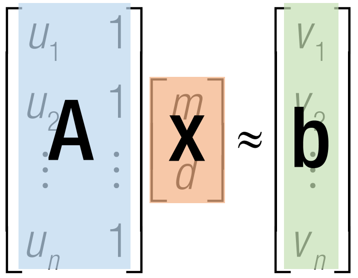
$$\frac{\partial e}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial e}{\partial x_1} & \cdots & \frac{\partial e}{\partial x_n} \end{bmatrix}$$

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

$$\frac{\partial \mathbf{c}^{\mathrm{T}} \mathbf{x}}{\partial \mathbf{x}} =$$

Line Fitting (Ax=b)

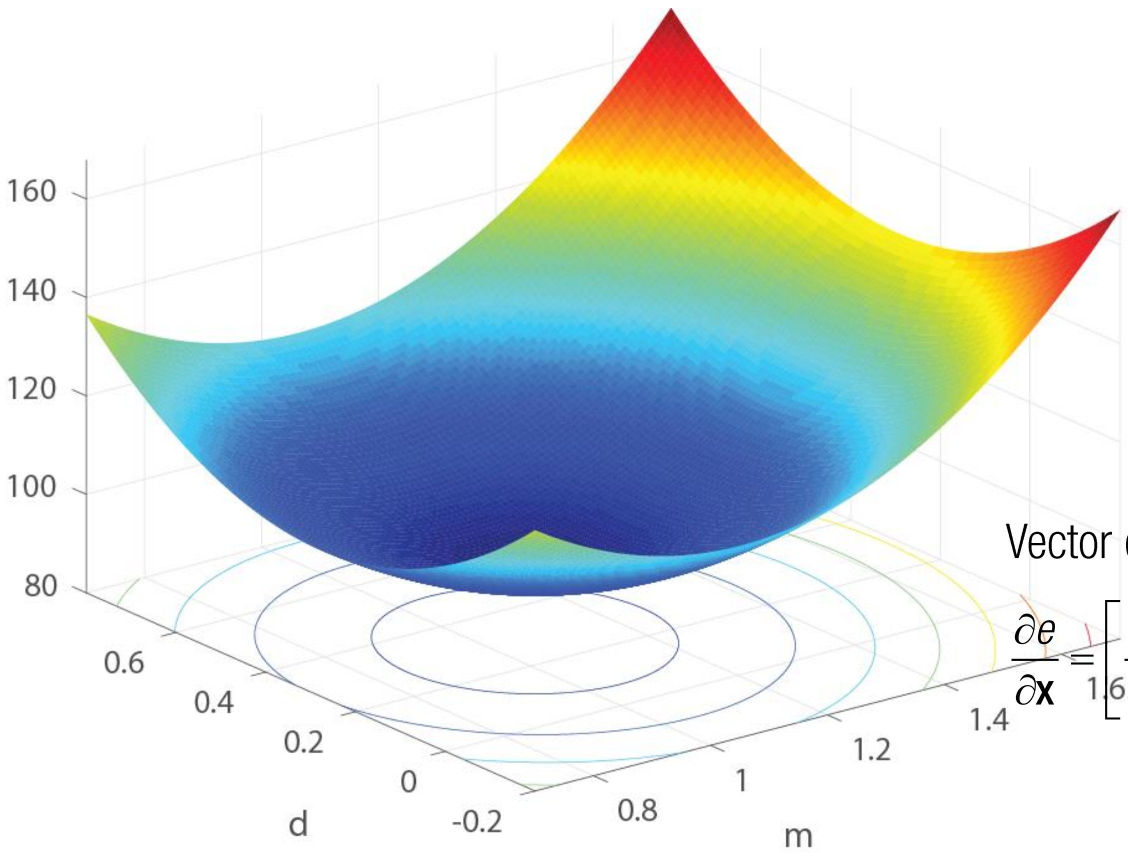
Total error: $E = \sum_{i=1}^{n} \left(v_i - (mu_i + d)\right)^2$



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

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

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



Vector derivative:

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

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

$$\frac{\partial \mathbf{c}^{\mathsf{T}}\mathbf{x}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \left(c_1 x_1 + \cdots + c_n x_n\right)$$

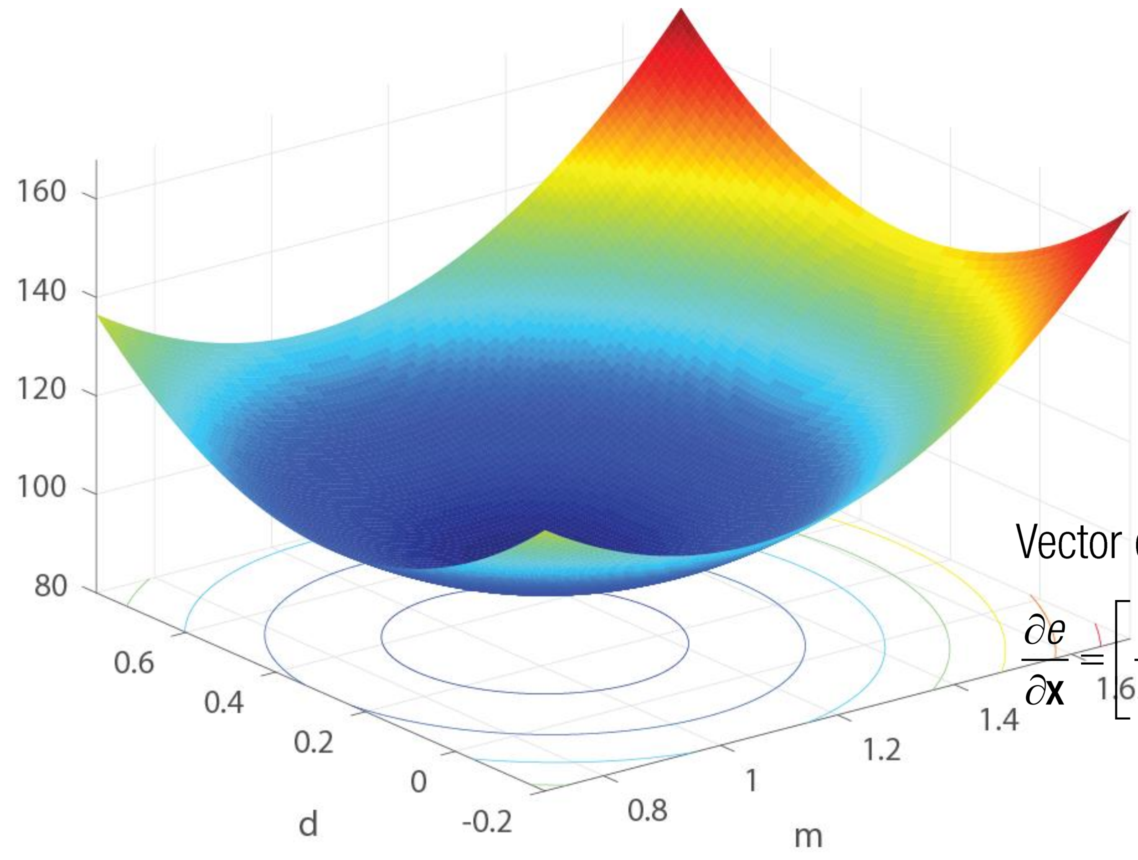
Line Fitting (Ax=b)

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

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

$$E = (\mathbf{Ax}-\mathbf{b})^{\mathsf{T}}(\mathbf{Ax}-\mathbf{b}) = \|\mathbf{Ax}-\mathbf{b}\|^2$$
$$= \mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{Ax} - 2\mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{b} + \mathbf{b}^{\mathsf{T}}\mathbf{b}$$

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



Vector derivative:

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

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

$$\frac{\partial \mathbf{c}^{\mathsf{T}}\mathbf{x}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}}(c_1x_1 + \cdots + c_nx_n)$$
$$= \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}$$

Line Fitting (Ax=b)

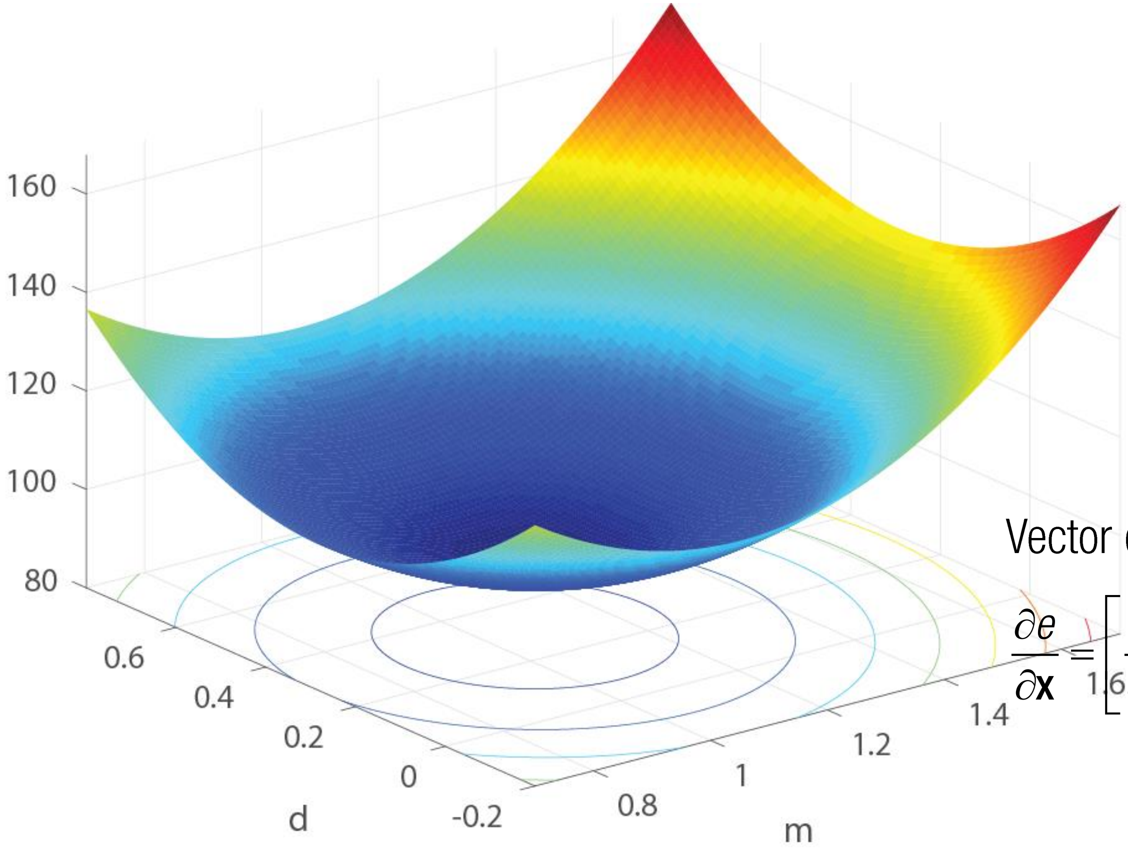
Total error: $E = \sum_{i=1}^{n} \left(v_i - (mu_i + d)\right)^2$

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

$$E = \left(\mathbf{Ax} - \mathbf{b}\right)^{\mathsf{T}} \left(\mathbf{Ax} - \mathbf{b}\right) = \left\|\mathbf{Ax} - \mathbf{b}\right\|^2$$

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

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



Vector derivative:

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

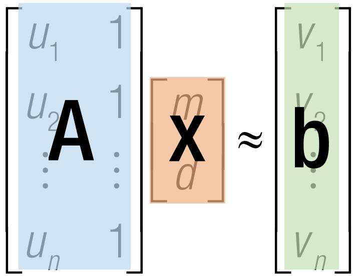
Ex) $e = \mathbf{c}^{\mathsf{T}}\mathbf{x} = \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

$$\frac{\partial \mathbf{c}^{\mathsf{T}}\mathbf{x}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}}\left(c_1 x_1 + \cdots + c_n x_n\right)$$

$$= \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}$$

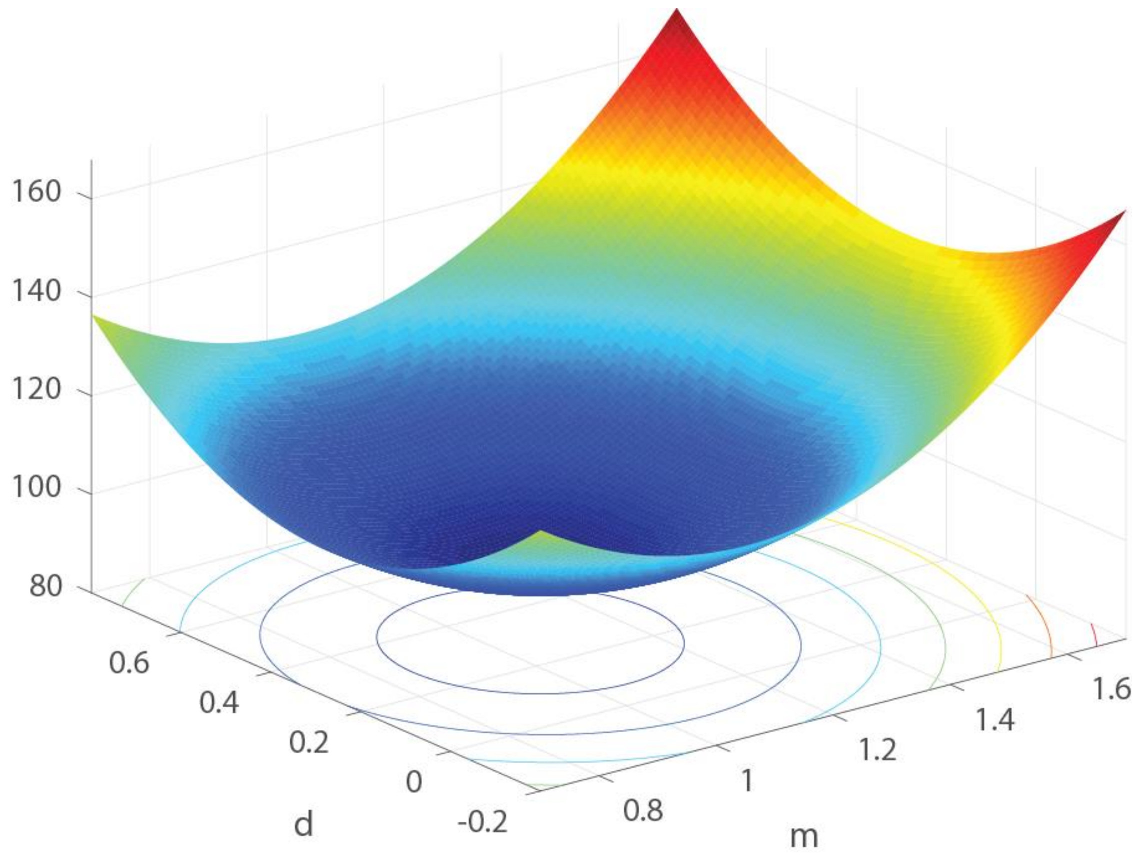
Line Fitting (Ax=b)

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



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

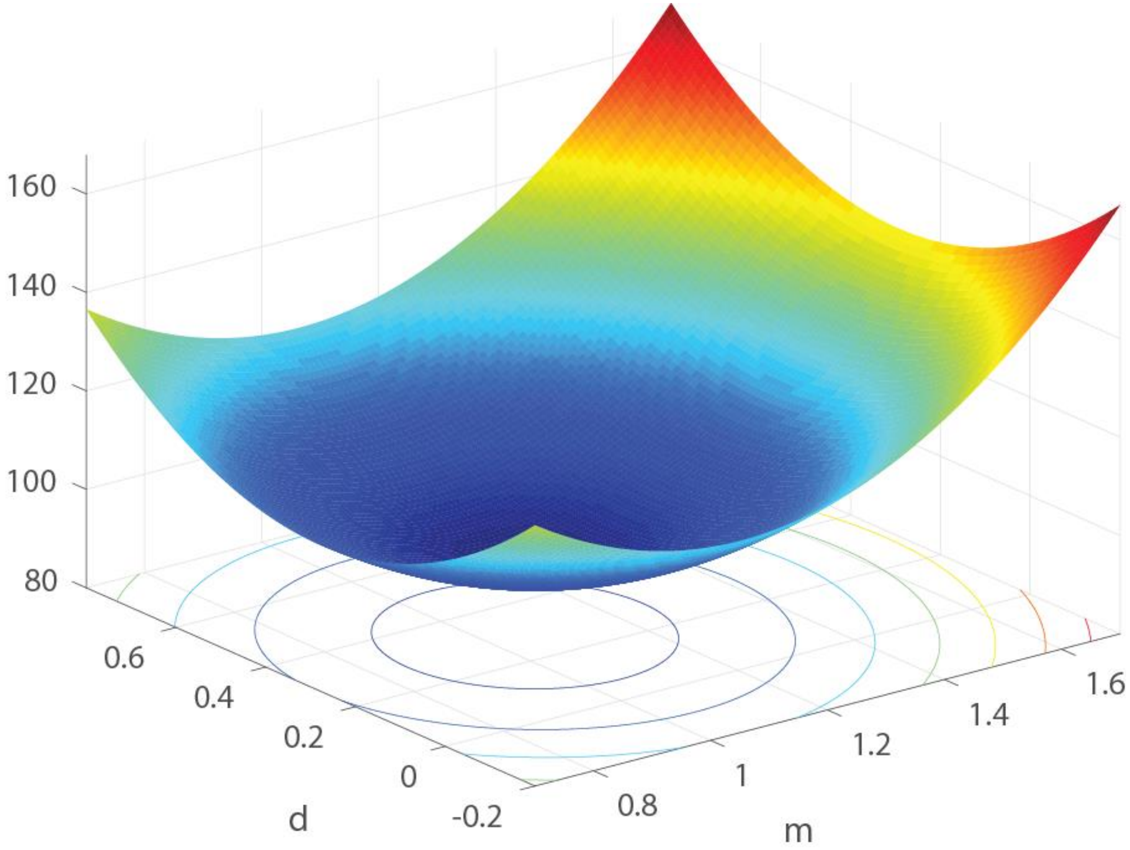
$$= \mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{Ax} - 2\mathbf{x}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{b} + \mathbf{b}^{\mathsf{T}}\mathbf{b}$$



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

$$\longrightarrow \mathbf{A}^{\mathsf{T}}\mathbf{Ax} = \mathbf{A}^{\mathsf{T}}\mathbf{b}$$

Line Fitting (Ax=b)



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

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

$$\mathbf{A}\quad \mathbf{x} \approx \mathbf{b}$$

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

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

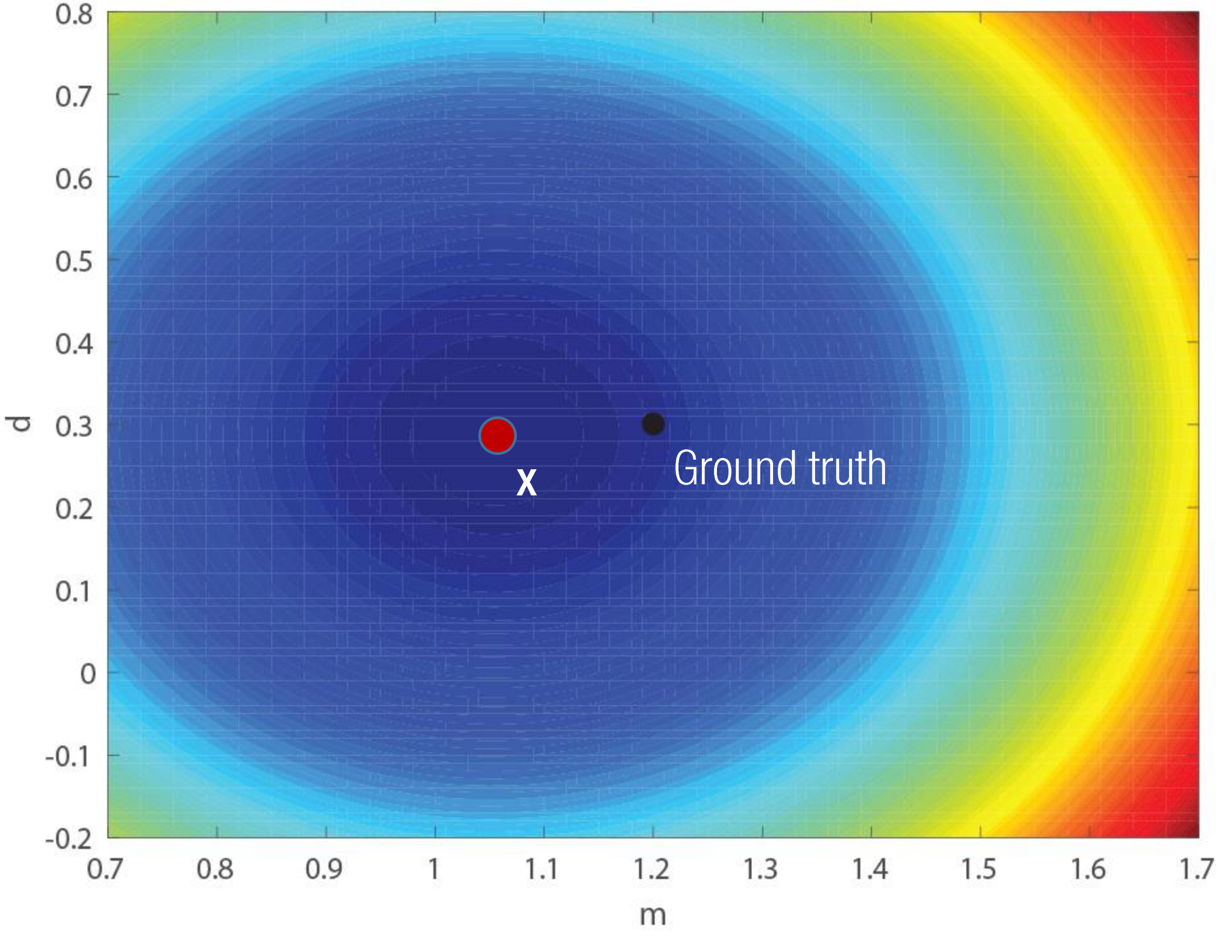
$$\frac{\partial E}{\partial \mathbf{x}}=2\mathbf{A}^{T}\mathbf{Ax}-2\mathbf{A}^{T}\mathbf{b}=\mathbf{0}$$

$$\longrightarrow \mathbf{A}^{T}\mathbf{Ax}=\mathbf{A}^{T}\mathbf{b}$$

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

Normal equation

Line Fitting (Ax=b)



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

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

$$\mathbf{A}\mathbf{x} \approx \mathbf{b}$$

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

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

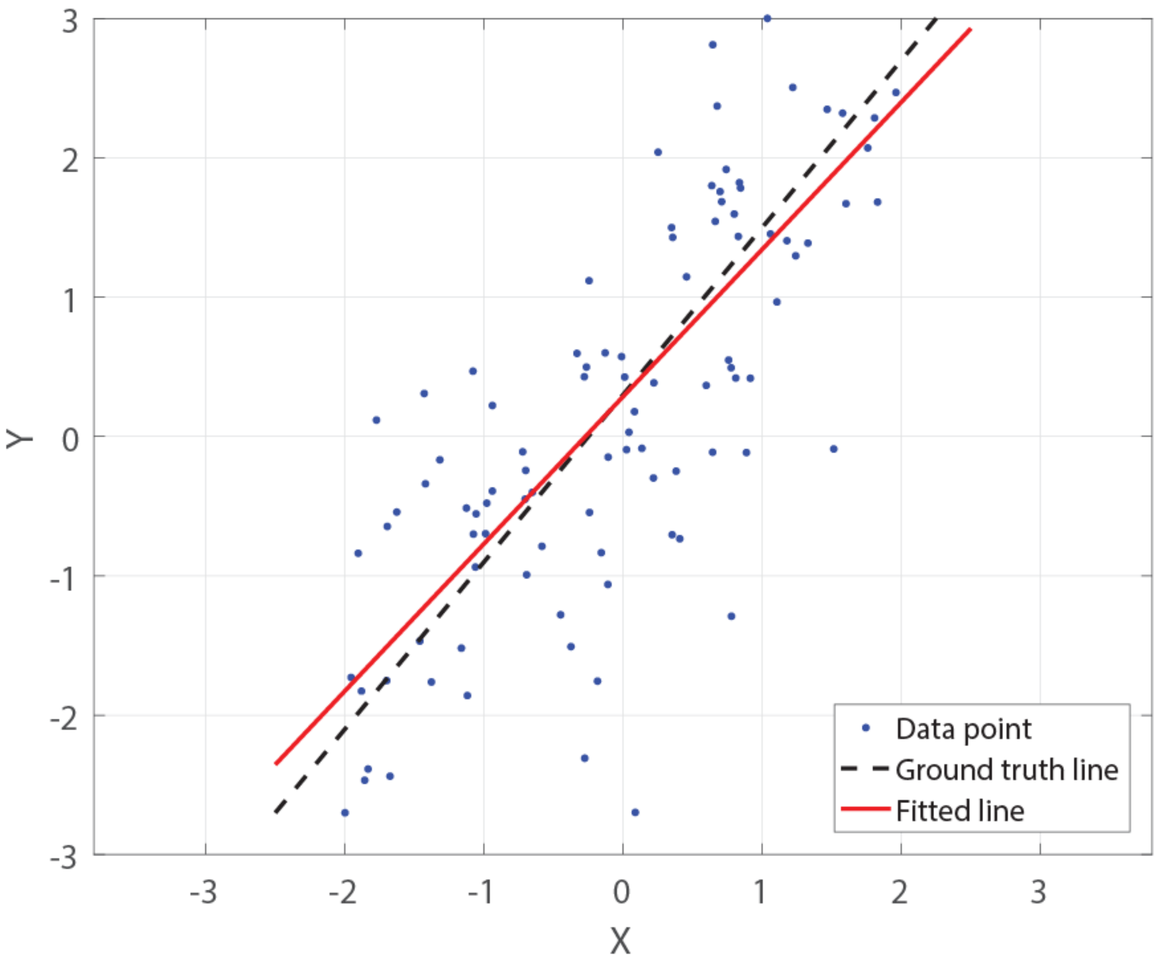
$$\frac{\partial E}{\partial \mathbf{x}} = 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b} = 0$$

$$\longrightarrow \mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

$$\mathbf{A}^T \; \mathbf{A} \; \mathbf{x} = \mathbf{A}^T \; \mathbf{b}$$

$$\mathbf{x} = \left[\mathbf{A}^T \; \mathbf{A}\right]^{-1} \mathbf{A}^T \; \mathbf{b}$$

Line Fitting (Ax=b)



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

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

$\mathbf{A}\,\mathbf{x} \approx \mathbf{b}$

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

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

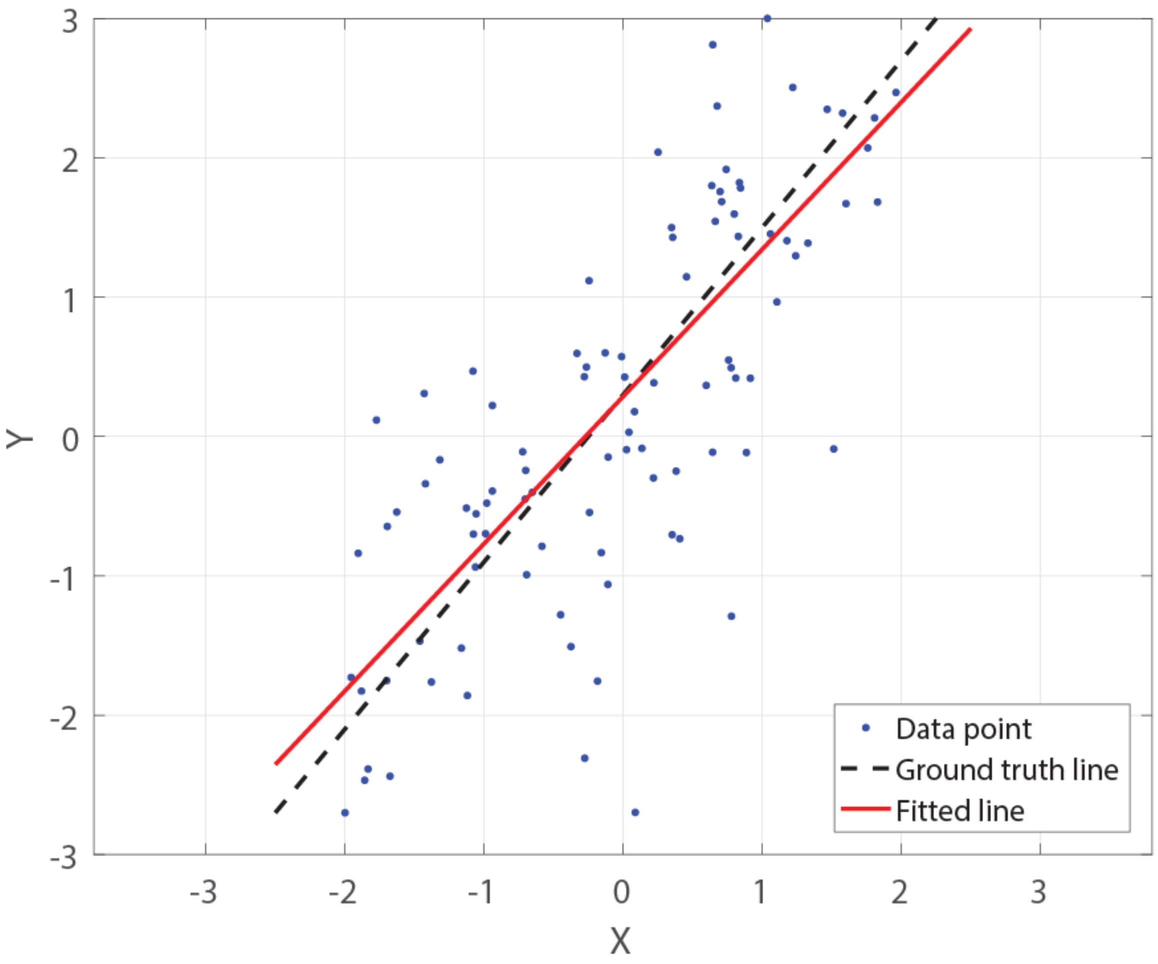
$$\frac{\partial E}{\partial \mathbf{x}} = 2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b} = \mathbf{0}$$

$$\longrightarrow \mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$$

$$\mathbf{A}^T \; \mathbf{A} \; \mathbf{x} = \mathbf{A}^T \; \mathbf{b}$$

$$\mathbf{x} = \left[\mathbf{A}^T \; \mathbf{A}\right]^{-1} \mathbf{A}^T \; \mathbf{b}$$

Line Fitting (Ax=b)



LineFitting.m

```
m = 1.2;
d = 0.3;

x = 4*(rand(100,1)-.5);
y = m*x + d + randn(size(x));

A = [x ones(size(x))];
b = y;

u = A\b;
```

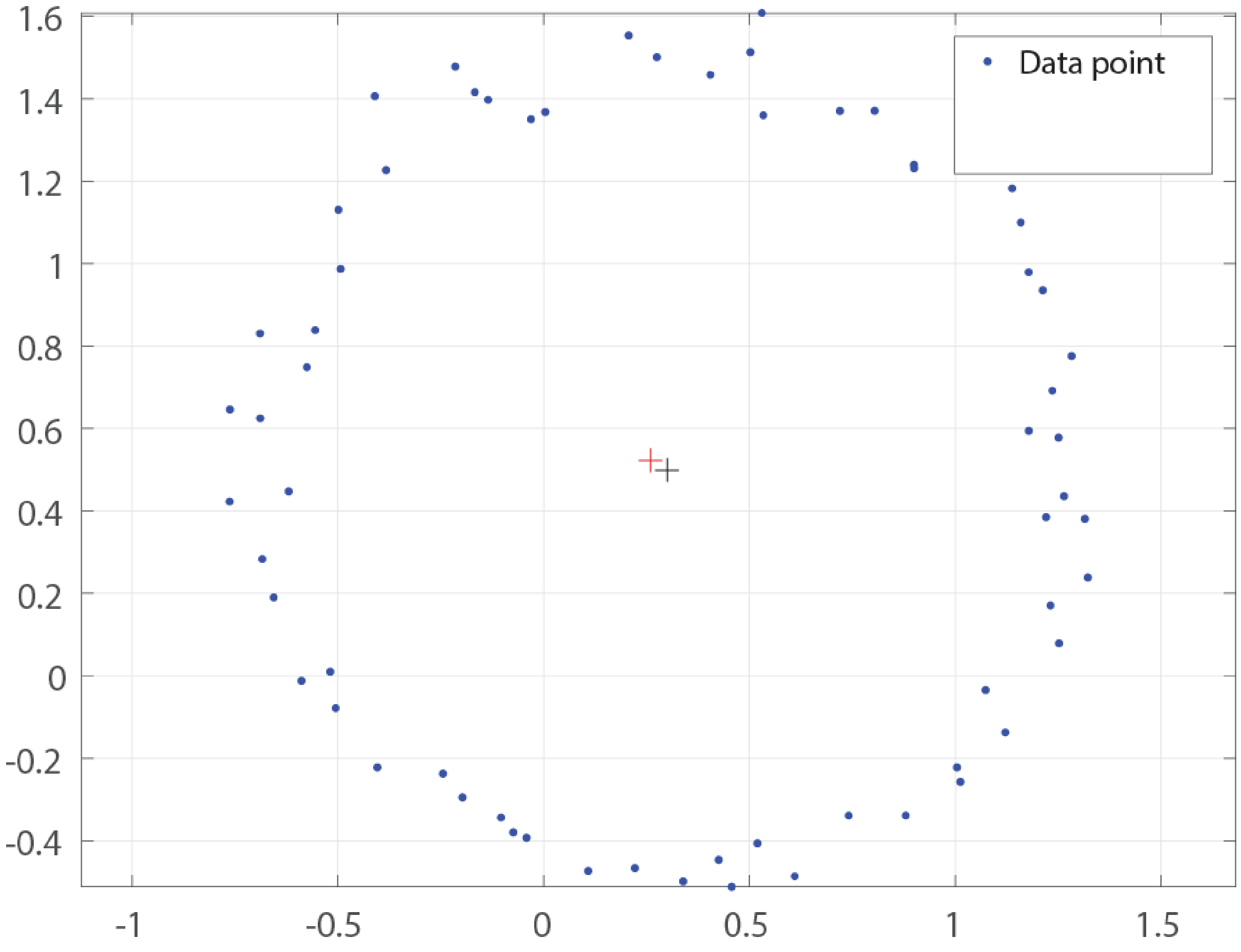
Ground truth

Random data point w/ Gaussian noise

$$\mathbf{x} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

\ backslush: solve linear least squares

Circle Fitting (Ax=b)



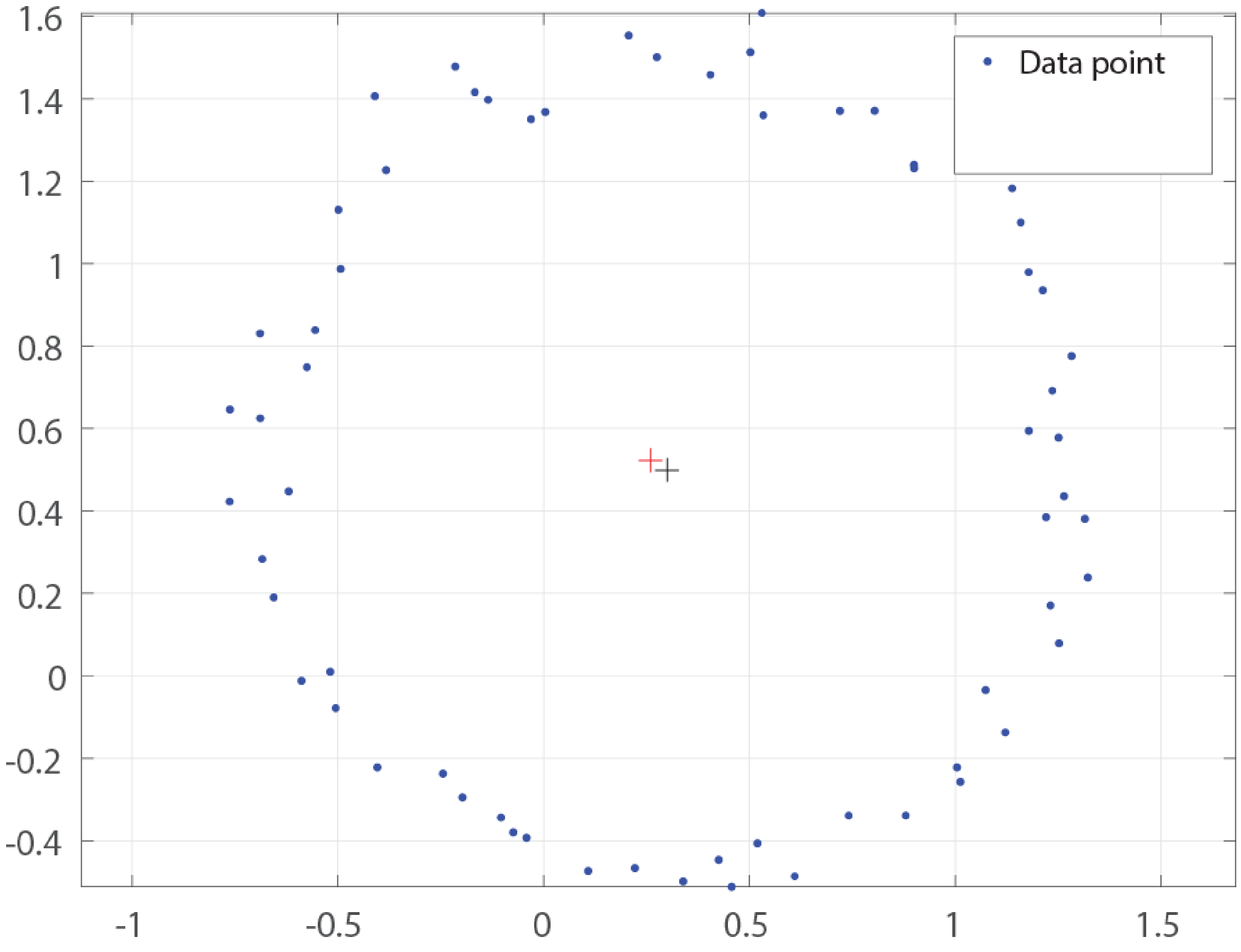
$$\left(x_1 - c_x\right)^2 + \left(y_1 - c_y\right)^2 = r^2$$

$$\vdots$$

$$\left(x_n - c_x\right)^2 + \left(y_n - c_y\right)^2 = r^2$$

Unkowns: c_x, c_y, r

Circle Fitting (Ax=b)

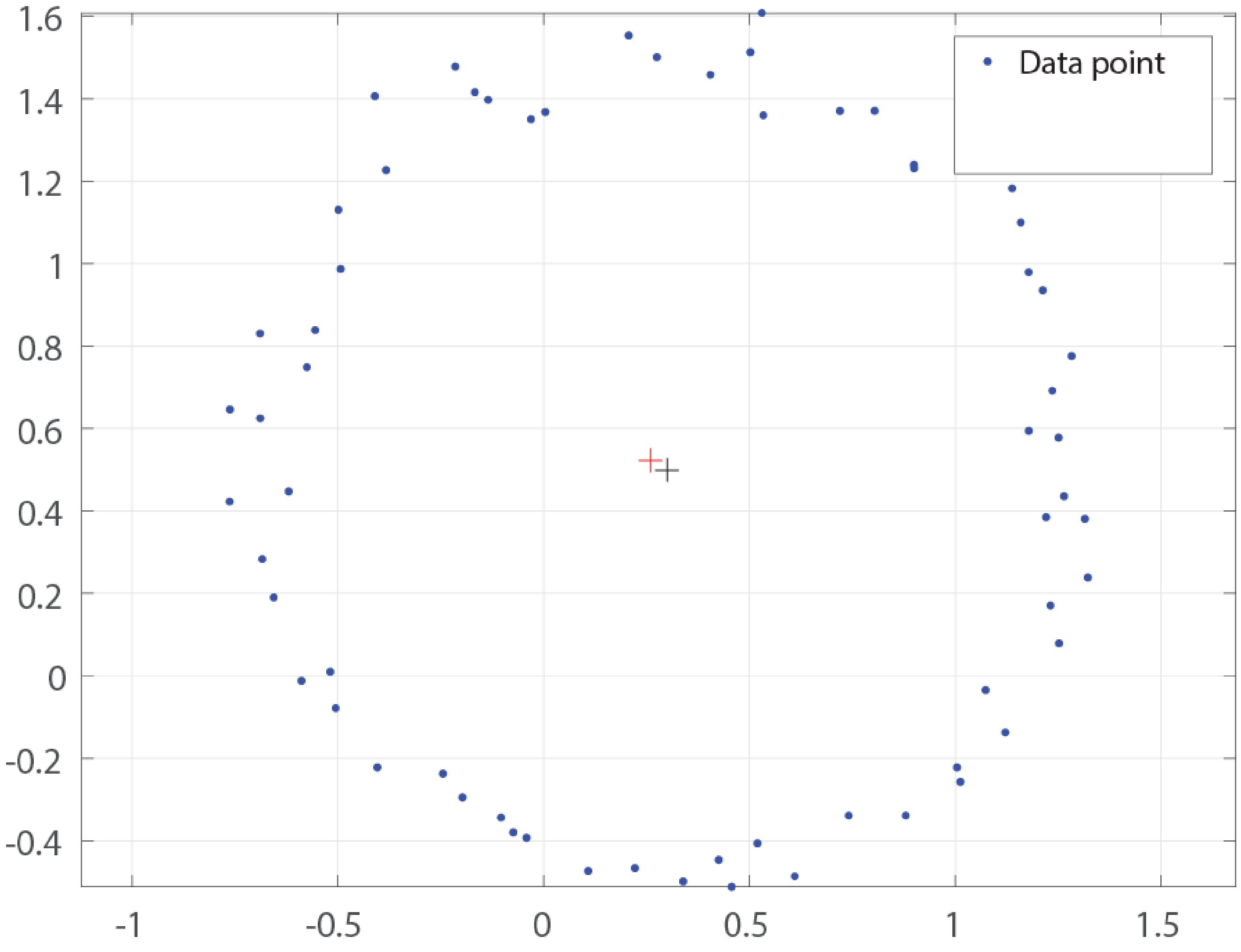


$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

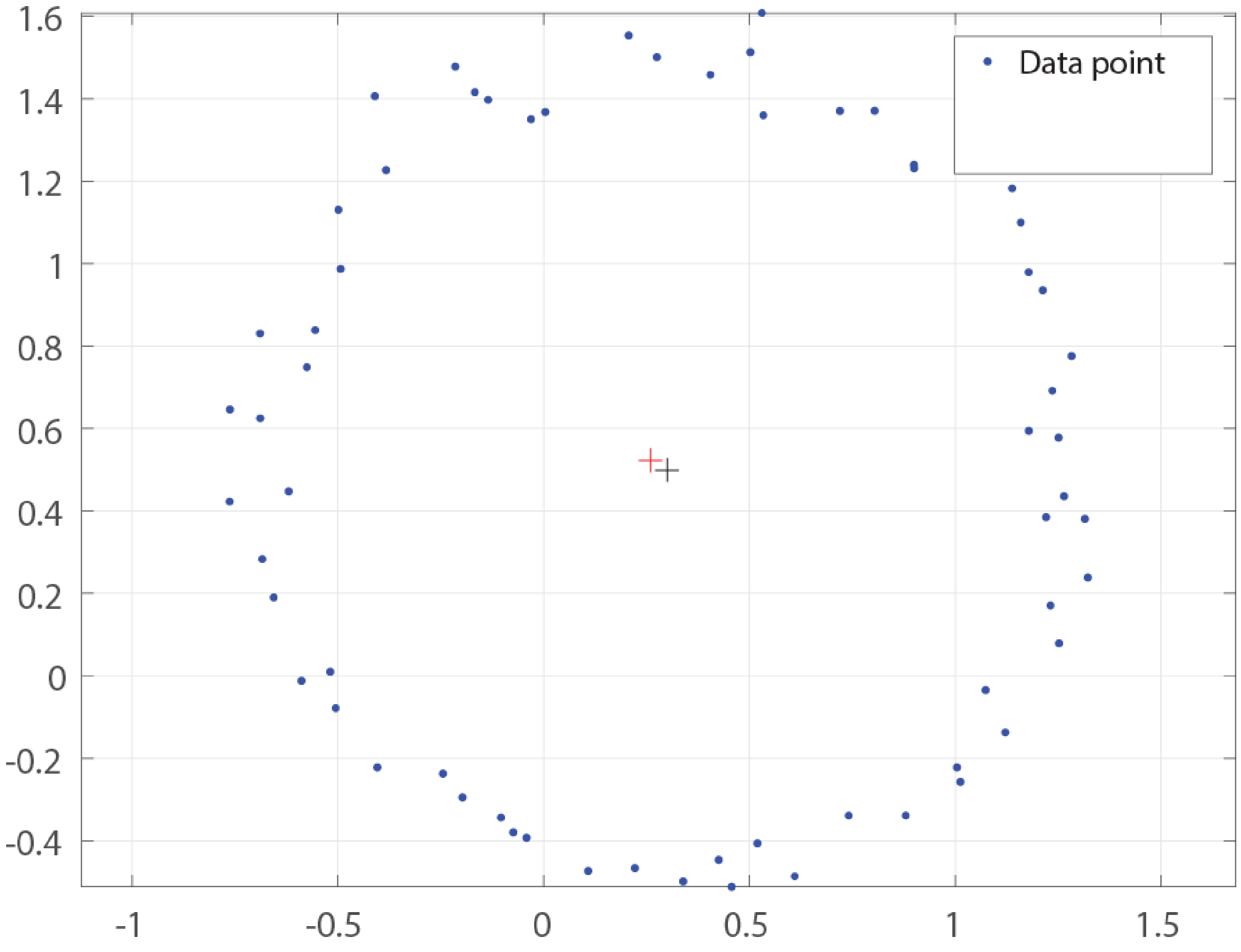
$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

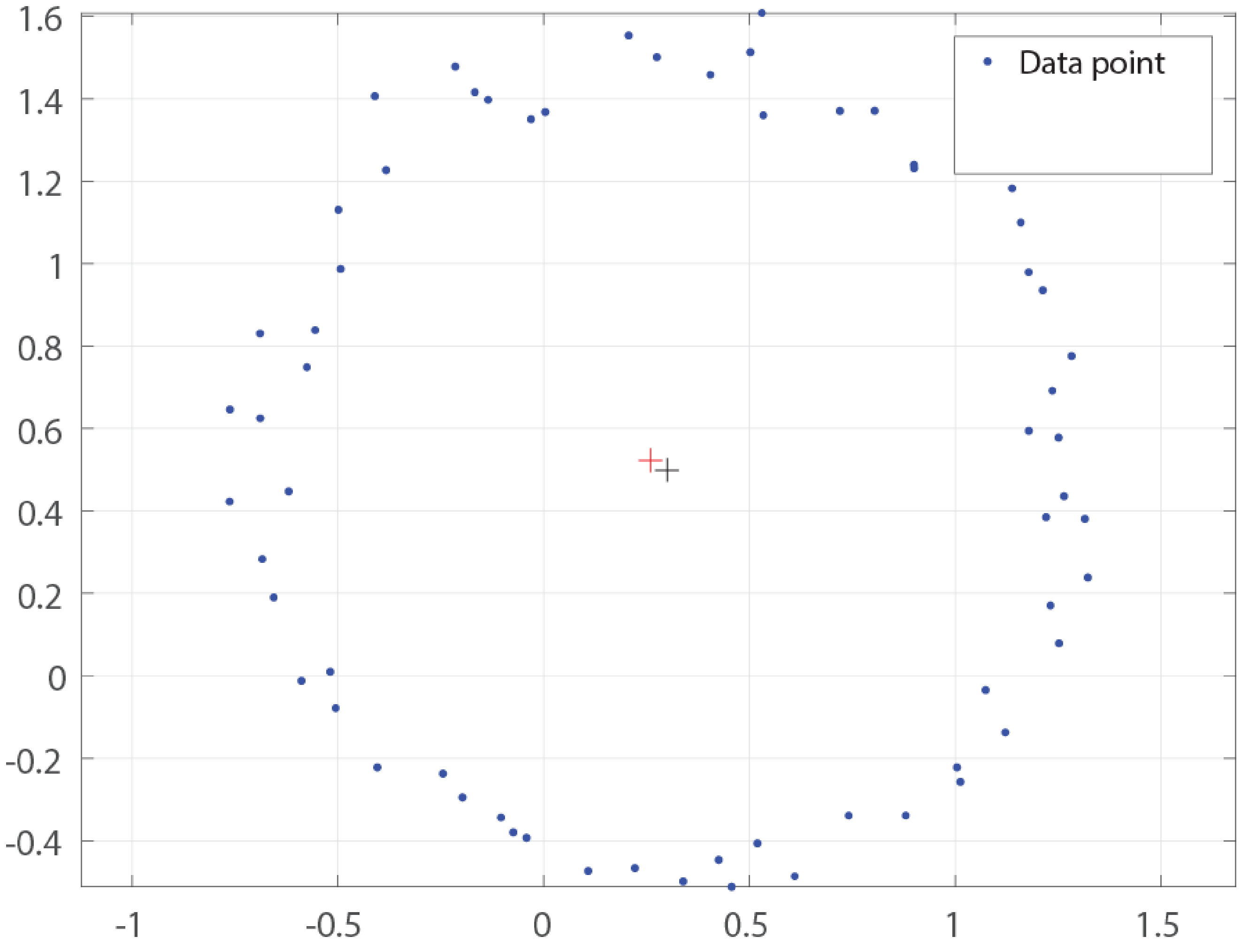
$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

$$\downarrow$$

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

Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

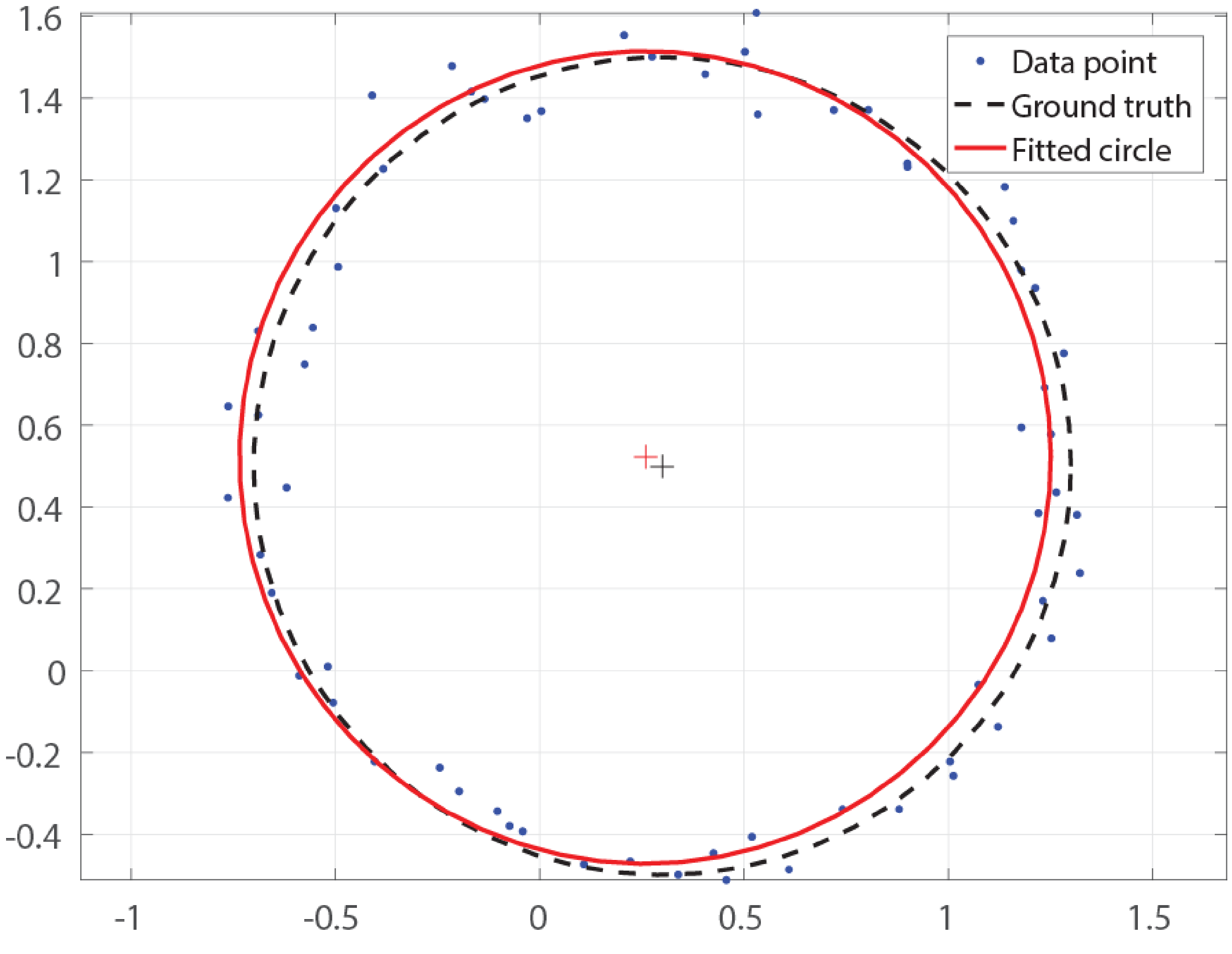
$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

$$\downarrow$$

$$\underbrace{\begin{bmatrix} 2\left(x_2 - x_1\right) & 2\left(y_2 - y_1\right) \\ \vdots & \vdots \\ 2\left(x_n - x_1\right) & 2\left(y_n - y_1\right) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_x \\ c_y \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \vdots \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}}_{\mathbf{b}}$$

Circle Fitting (Ax=b)



$$x_1^2 - 2x_1 c_x + c_x^2 + y_1^2 - 2y_1 c_y + c_y^2 = r^2$$

$$\vdots$$

$$x_n^2 - 2x_n c_x + c_x^2 + y_n^2 - 2y_n c_y + c_y^2 = r^2$$

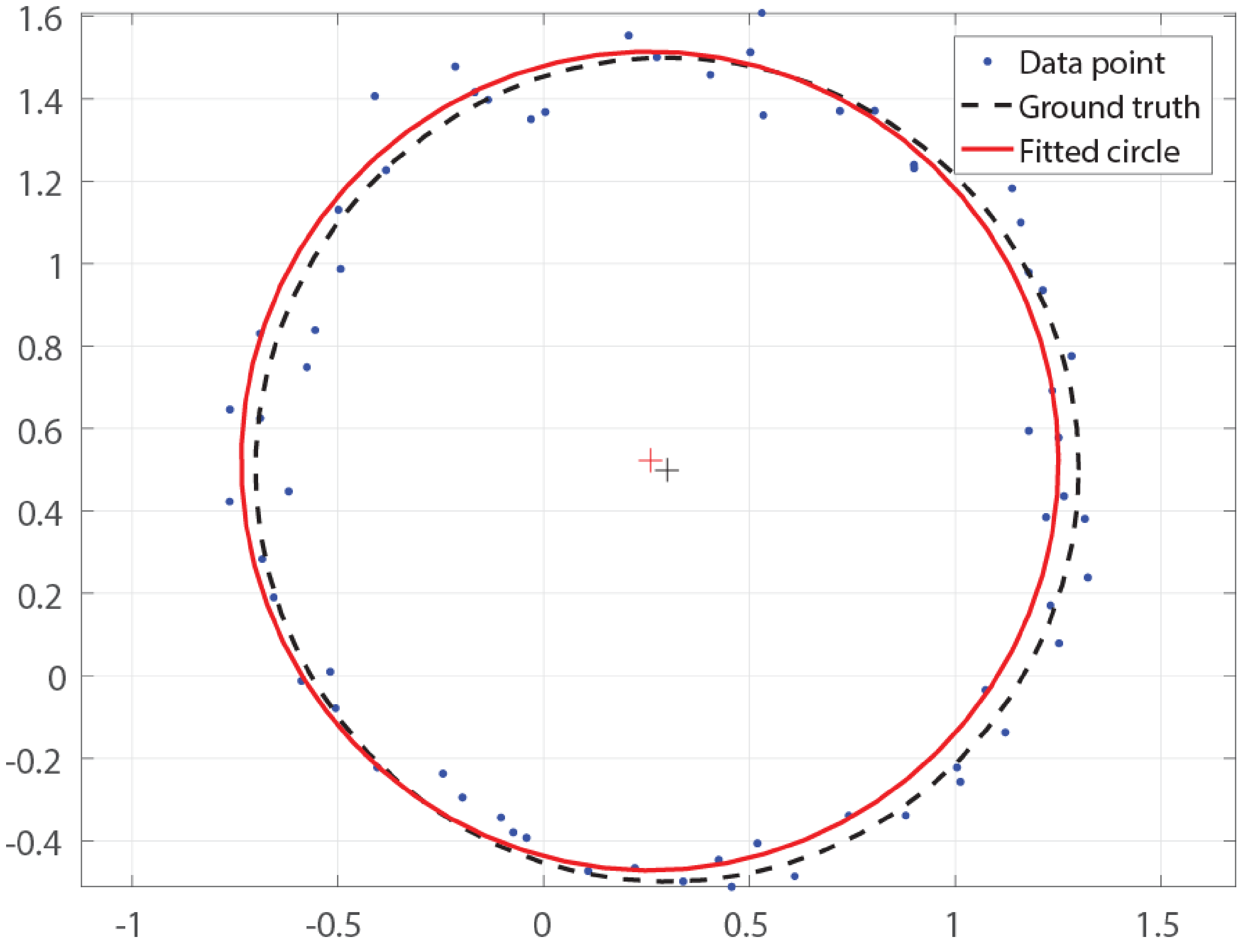
$$\downarrow$$

$$x_i^2 - x_1^2 - 2c_x\left(x_i - x_1\right) + y_i^2 - y_1^2 - 2\left(y_i - y_1\right)c_y = 0$$

$$\downarrow$$

$$\underbrace{\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ \vdots & \vdots \\ 2(x_n - x_1) & 2(y_n - y_1) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} c_x \\ c_y \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ \vdots \\ x_n^2 - x_1^2 + y_n^2 - y_1^2 \end{bmatrix}}_{\mathbf{b}}$$

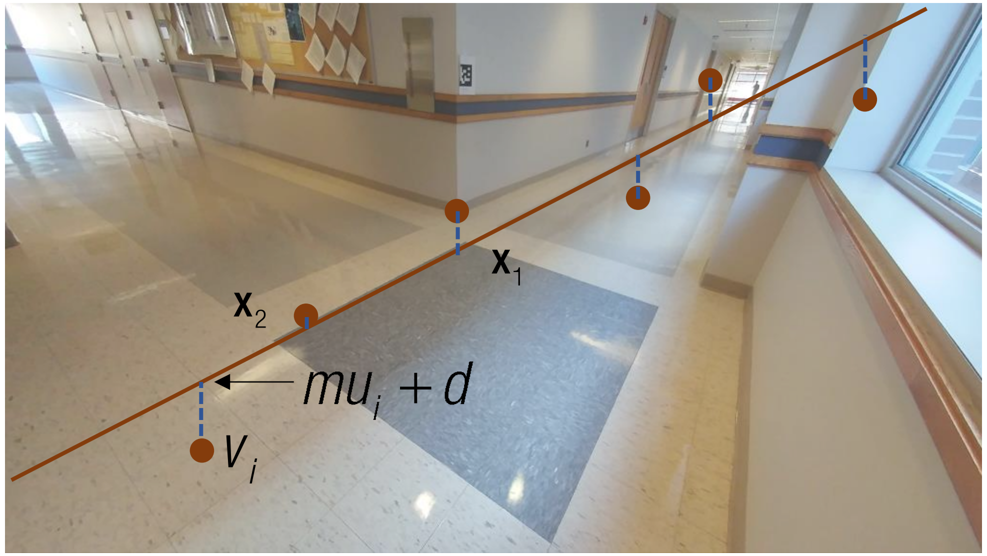
Circle Fitting (Ax=b)



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: $(u_1,v_1),(u_2,v_2),\cdots,(u_n,v_n)$

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

slope y-intercept

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

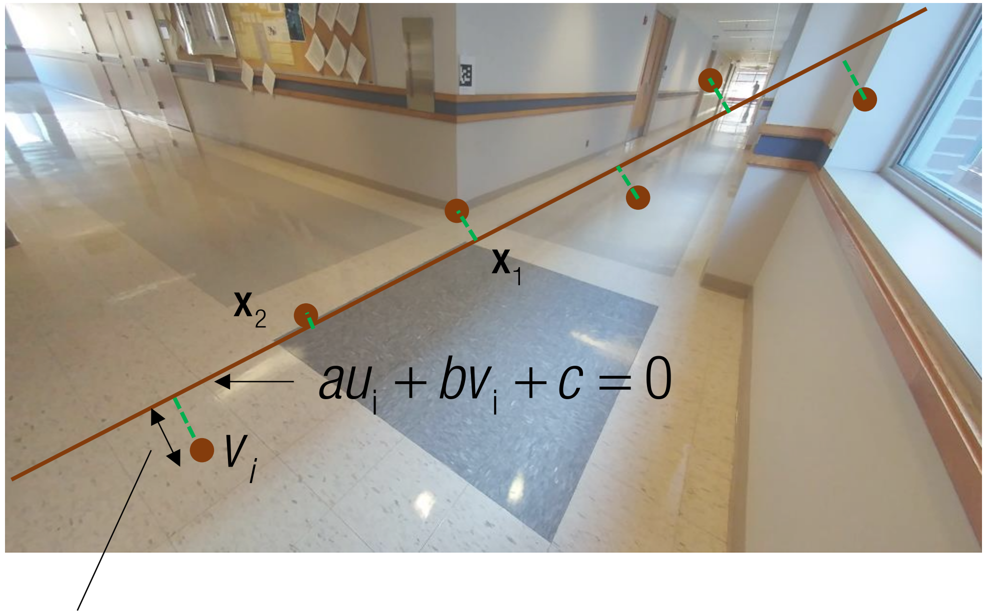
$$\mathbf{Ax} = \mathbf{b}$$

What is different?

Line Fitting (Ax=0)

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

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



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

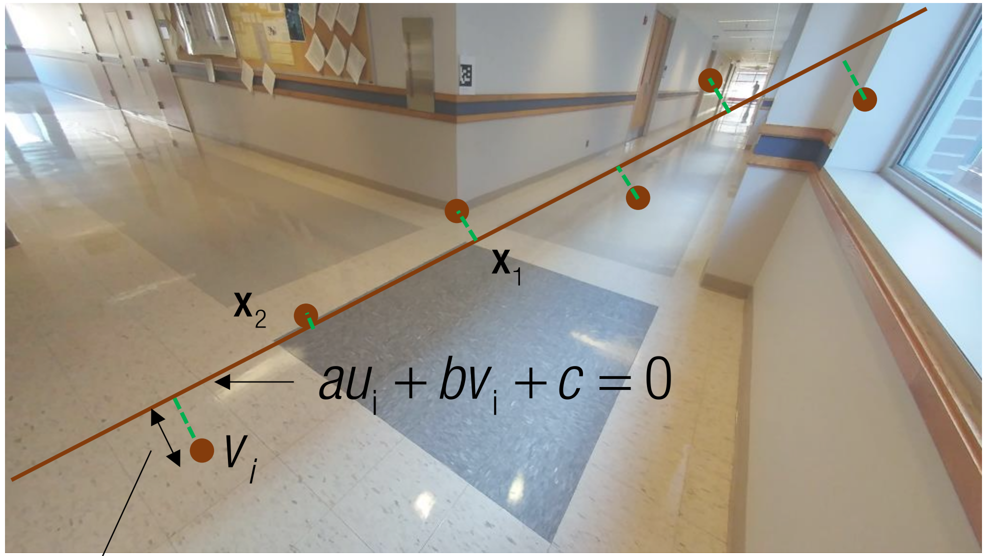
$$\begin{aligned} au_1 + bv_1 + c &\approx 0 \\ au_2 + bv_2 + c &\approx 0 \\ &\vdots \\ au_n + bv_n + c &\approx 0 \end{aligned} \quad \longrightarrow \quad \begin{aligned} v_1 &\approx mu_1 + d \\ v_2 &\approx mu_2 + d \\ &\vdots \\ v_n &\approx mu_n + d \end{aligned}$$

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

Line Fitting (Ax=0)

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

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



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

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

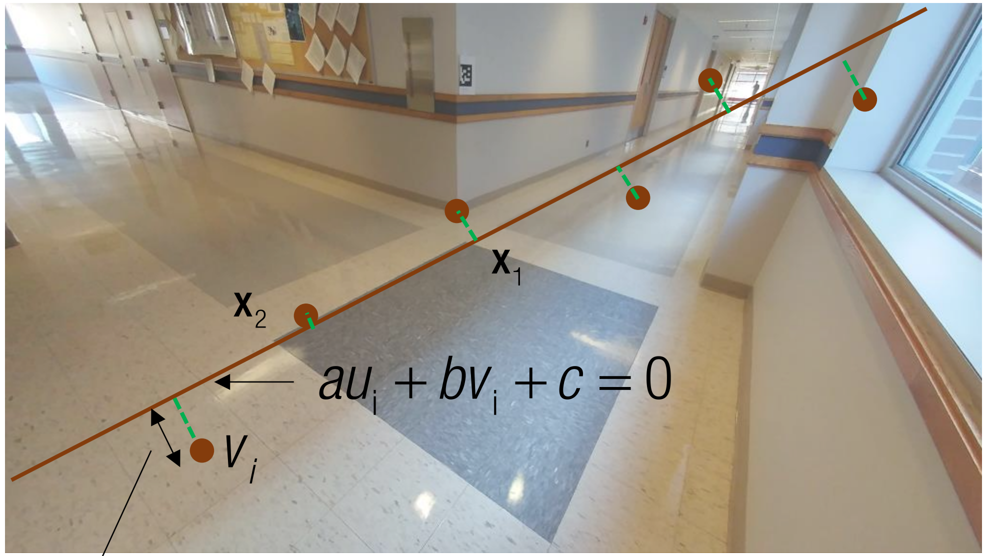
$\mathbf{Ax} = \mathbf{0}$ $\qquad$ $\mathbf{Ax} = \mathbf{b}$

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

Line Fitting (Ax=0)

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

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



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

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

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

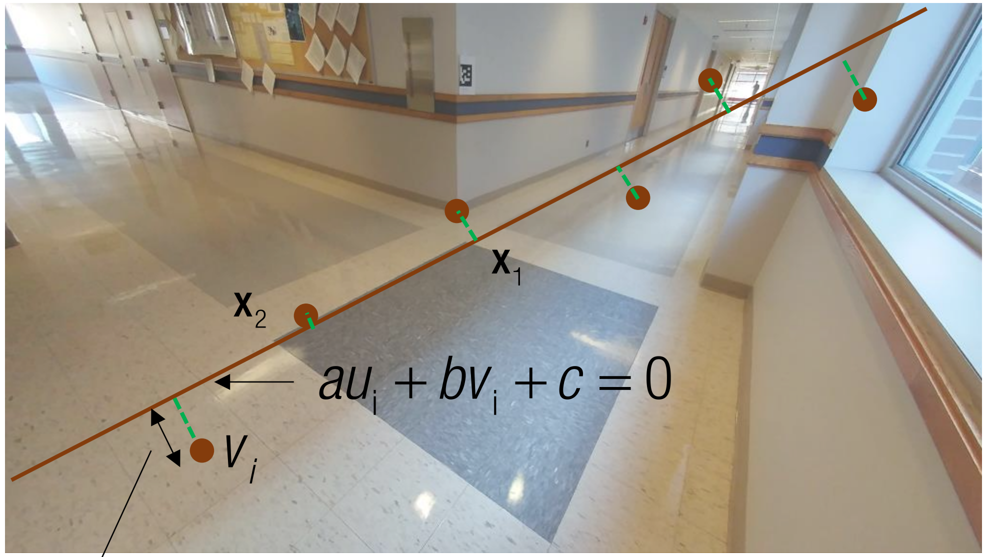
Trivial solution: $\mathbf{x} = \mathbf{0}$

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

Line Fitting (Ax=0)

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

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



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

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

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

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

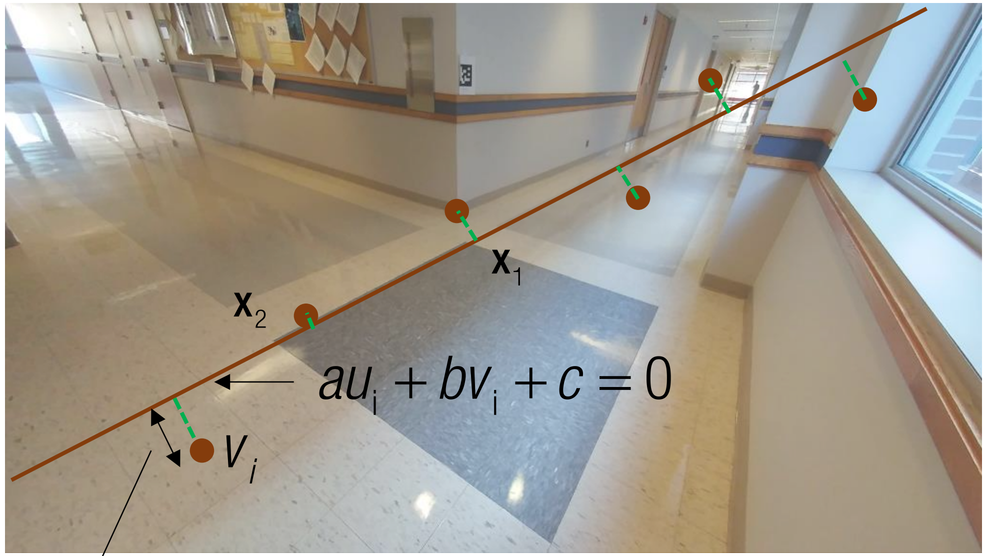
$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2 \qquad \text{subject to} \quad \|\mathbf{x}\| = 1$$

Condition to avoid the trivial solution

Line Fitting (Ax=0)

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

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



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

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

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

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

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

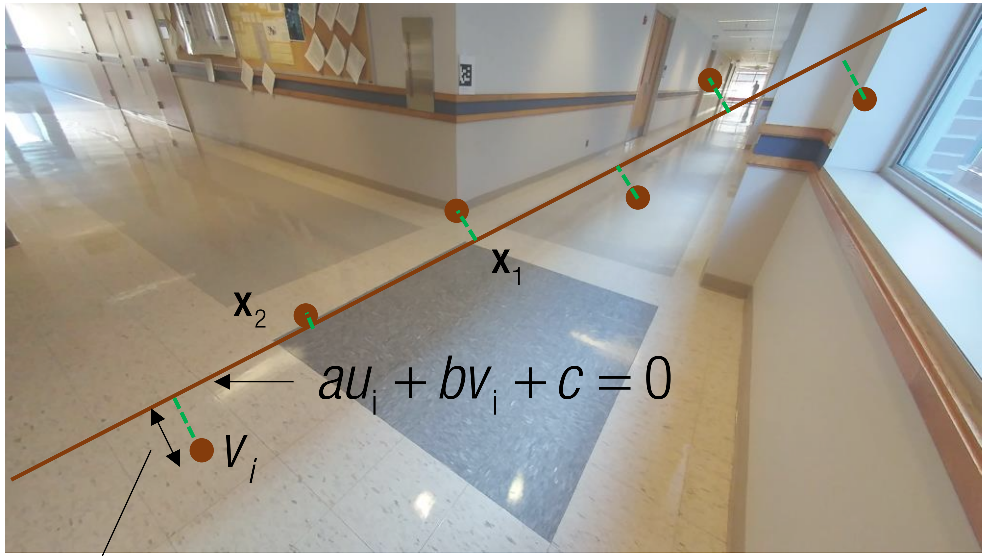
Condition to avoid the trivial solution

How to solve?

Line Fitting (Ax=0)

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

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



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

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

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

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

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

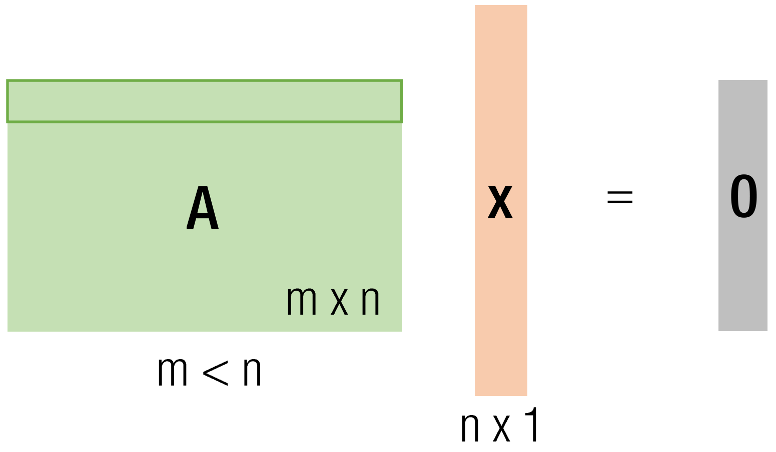
Condition to avoid the trivial solution

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

Nullspace

\# eqs < # unknowns

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

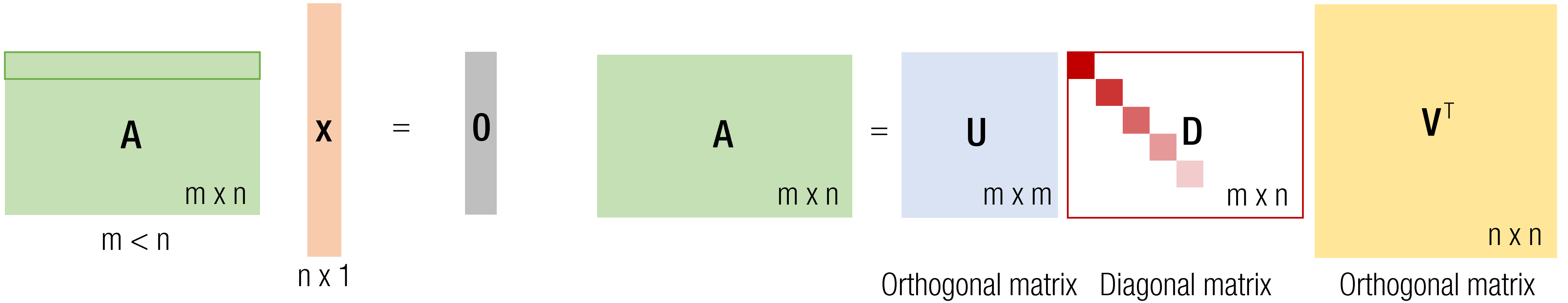


$\longrightarrow \mathbf{x} = \text{null}(\mathbf{A})$

Singular Value Decomposition (SVD)

eqs < # unknowns

There exist at least n-m vectors that makes **Ax**=**0**.

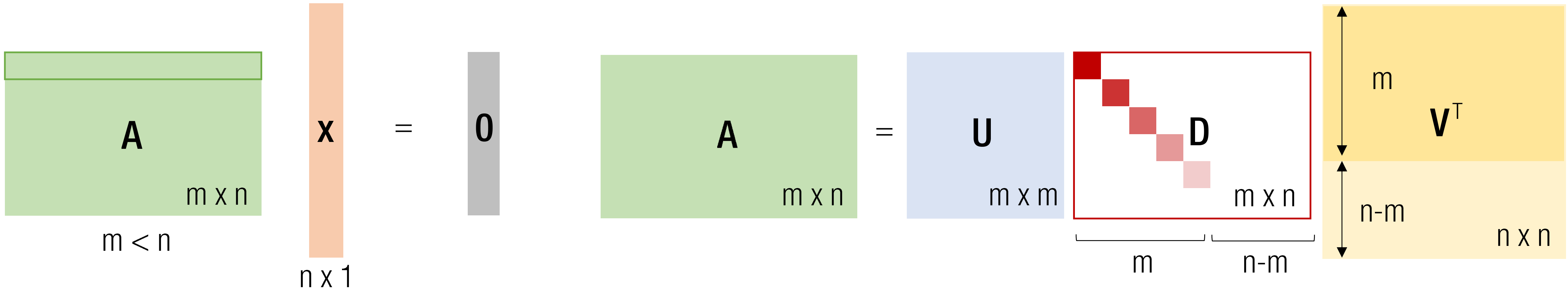


$\longrightarrow \mathbf{x} = \text{null}(\mathbf{A})$

Singular Value Decomposition (SVD)

\# eqs < # unknowns

There exist at least n-m vectors that makes **Ax**=**0**.

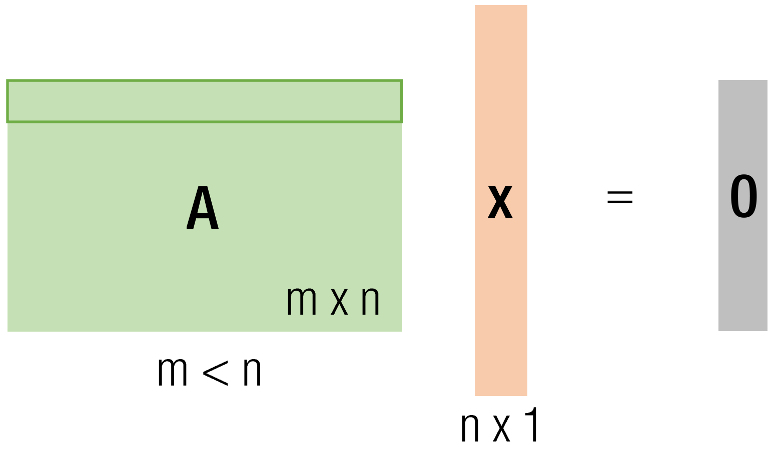


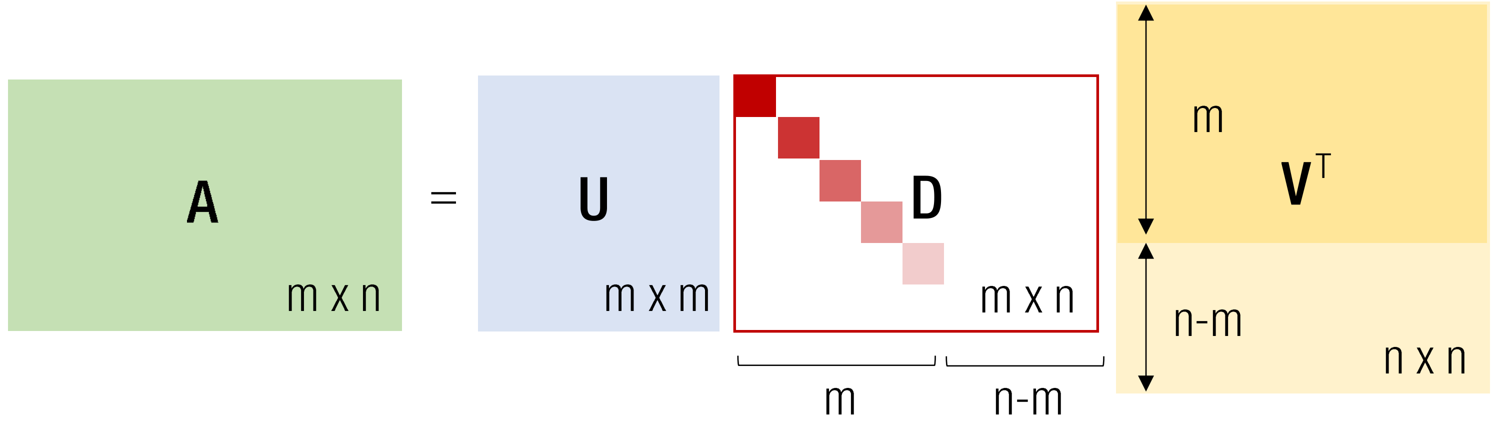
$$\longrightarrow \mathbf{x} = \text{null}(\mathbf{A})$$

Singular Value Decomposition (SVD)

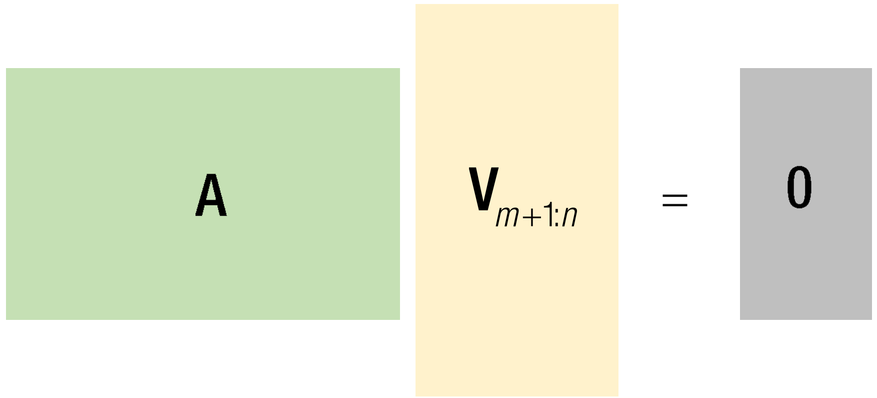
\# eqs < # unknowns

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





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

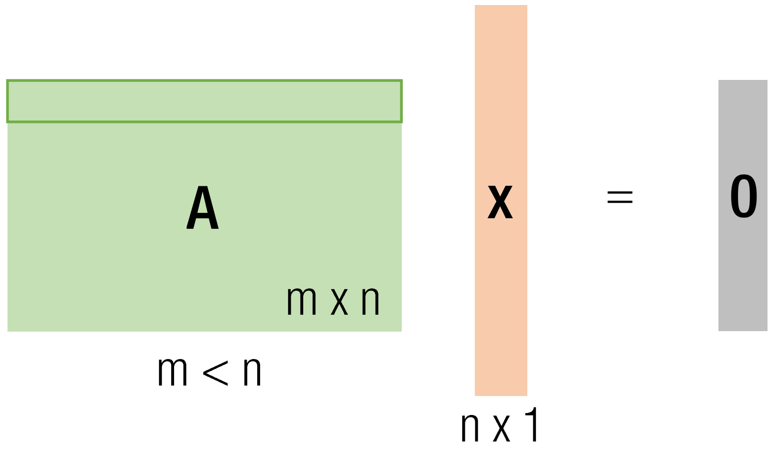


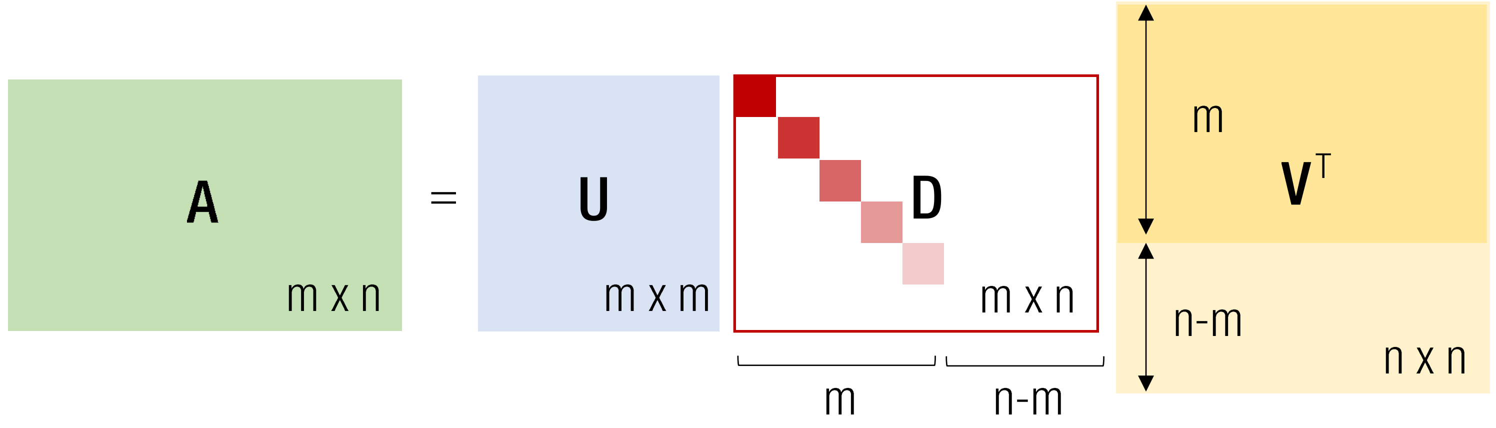
$$\mathbf{V}_{m+1:n} = \text{null}(\mathbf{A})$$

Singular Value Decomposition (SVD)

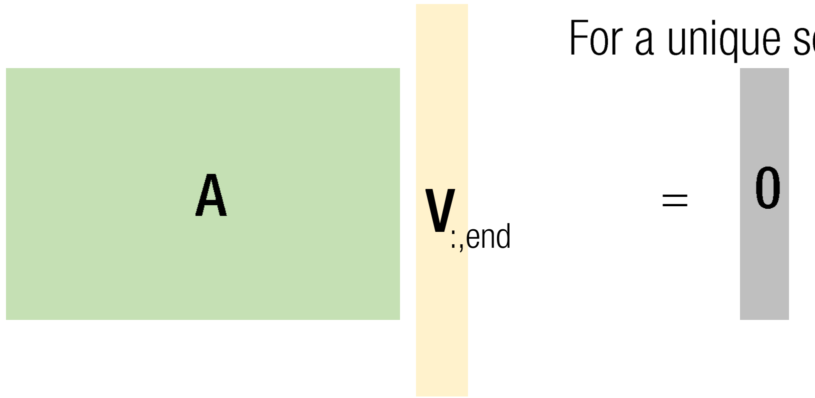
eqs < # unknowns

There exist at least n-m vectors that makes **Ax**=**0**.





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



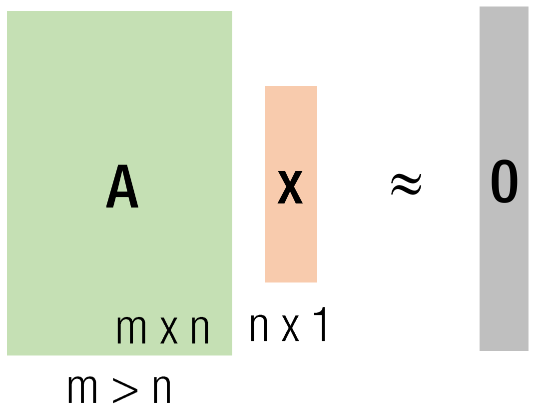
For a unique solution, **A** should be m x (m+1)

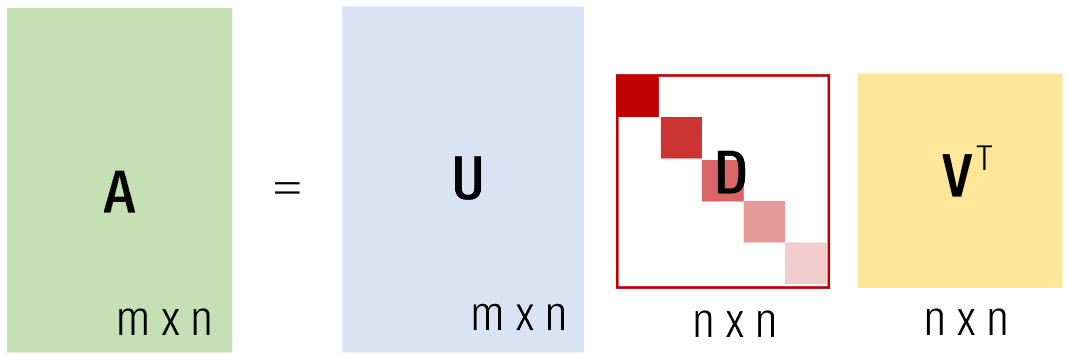
$$\mathbf{V}_{:end} = \text{null}(\mathbf{A})$$

Singular Value Decomposition (SVD)

eqs > # unknowns

There exist no nullspace of **A**.



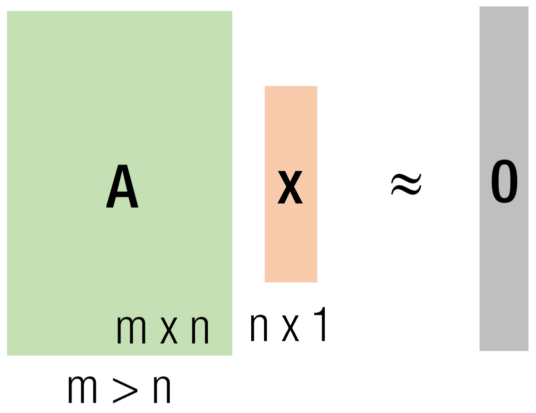


$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2 \text{ subject to } \|\mathbf{x}\| = 1$$

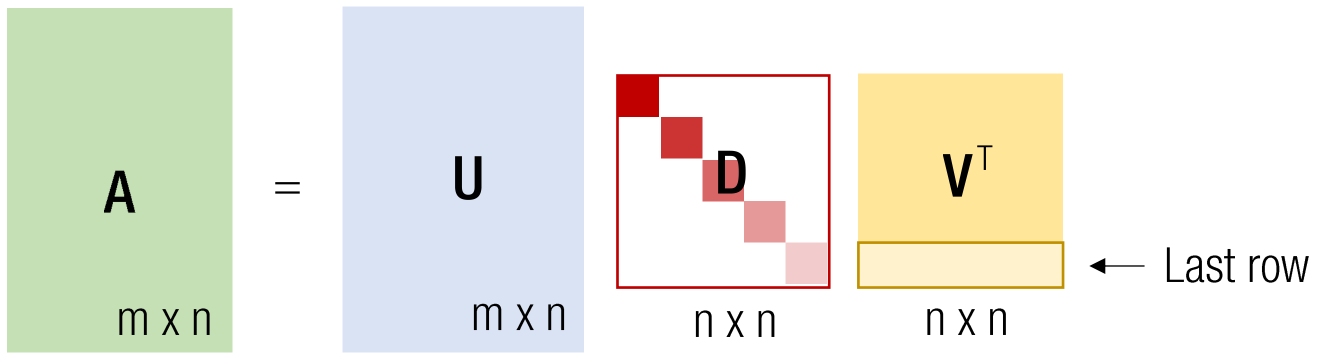
Singular Value Decomposition (SVD)

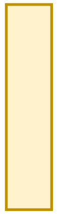
eqs > # unknowns

There exist no nullspace of **A**.

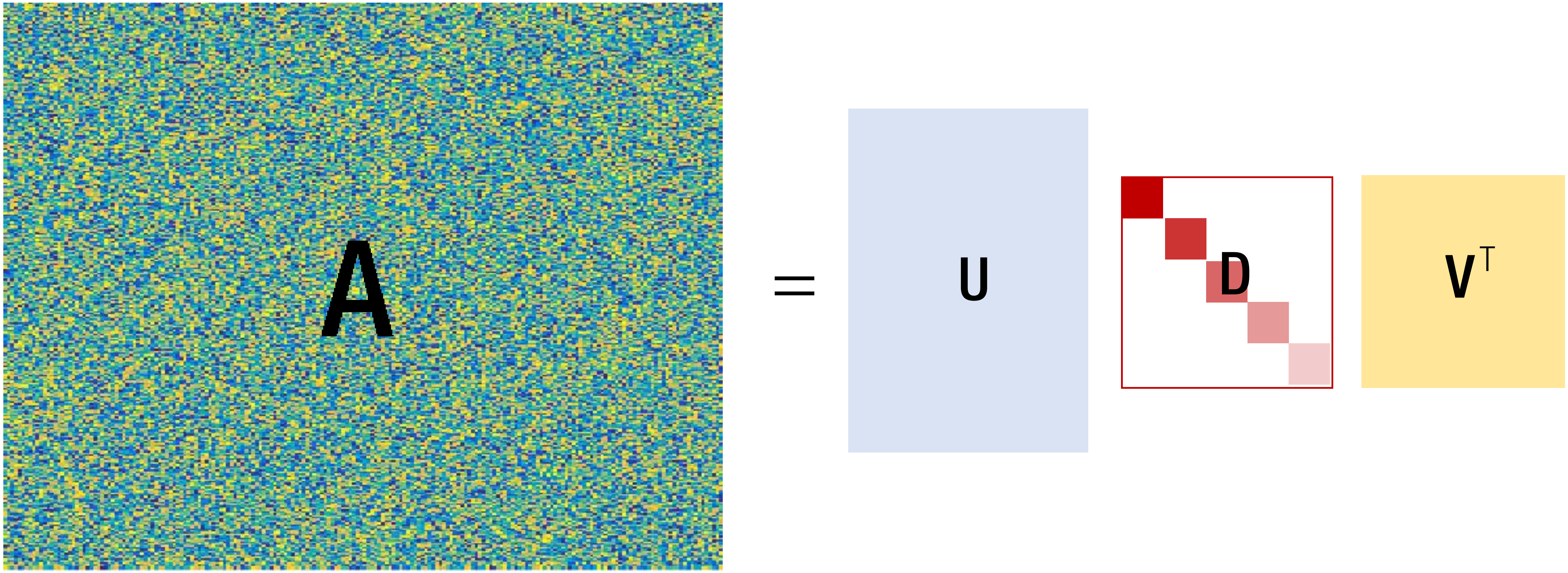


$$\underset{\mathbf{x}}{\text{minimize}} \|\mathbf{Ax}\|^2 \text{ subject to } \|\mathbf{x}\| = 1$$



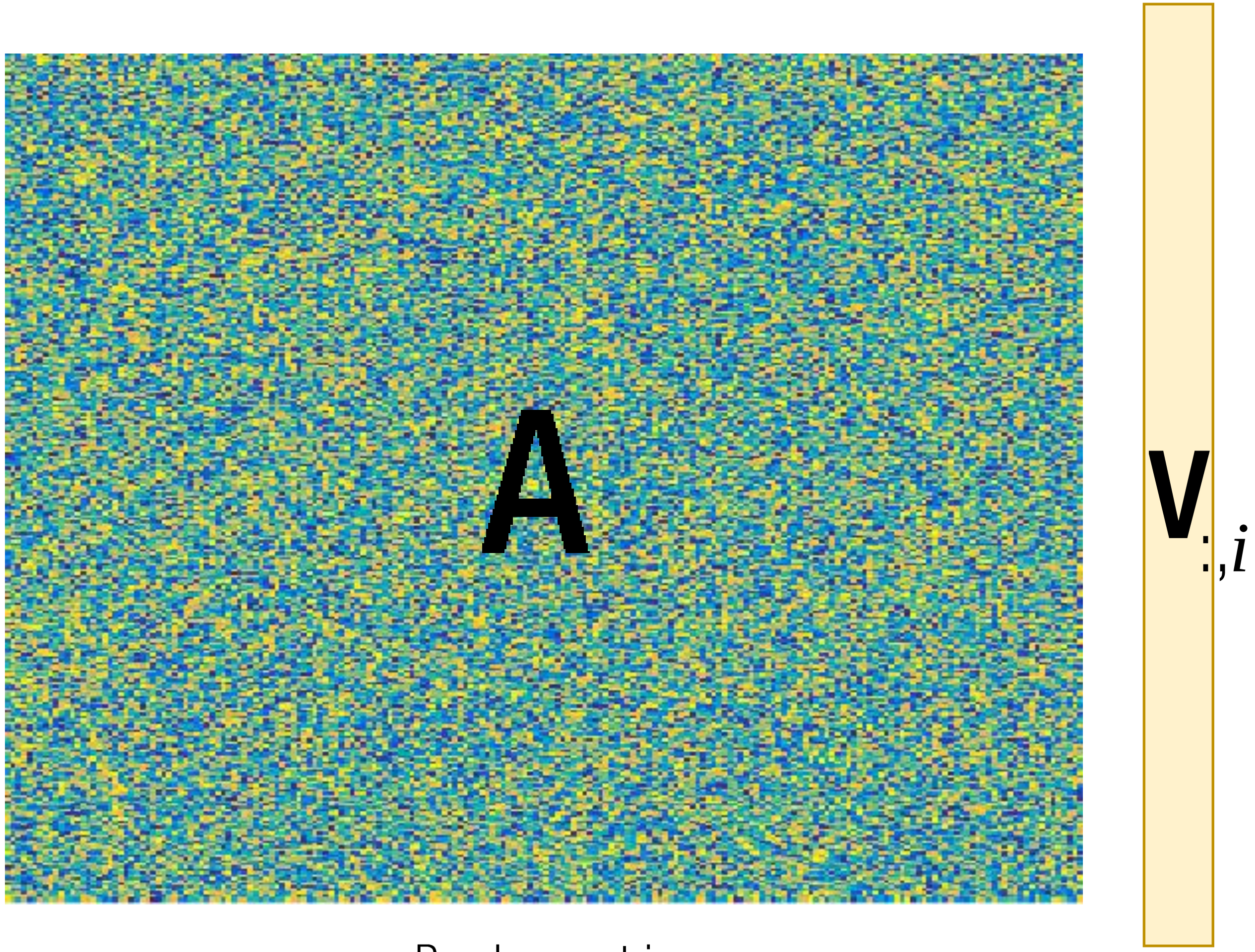
Approximated nullspace of **A**:  $\mathbf{V}_{:,end}$

Random Matrix SVD



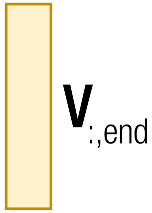
Random matrix

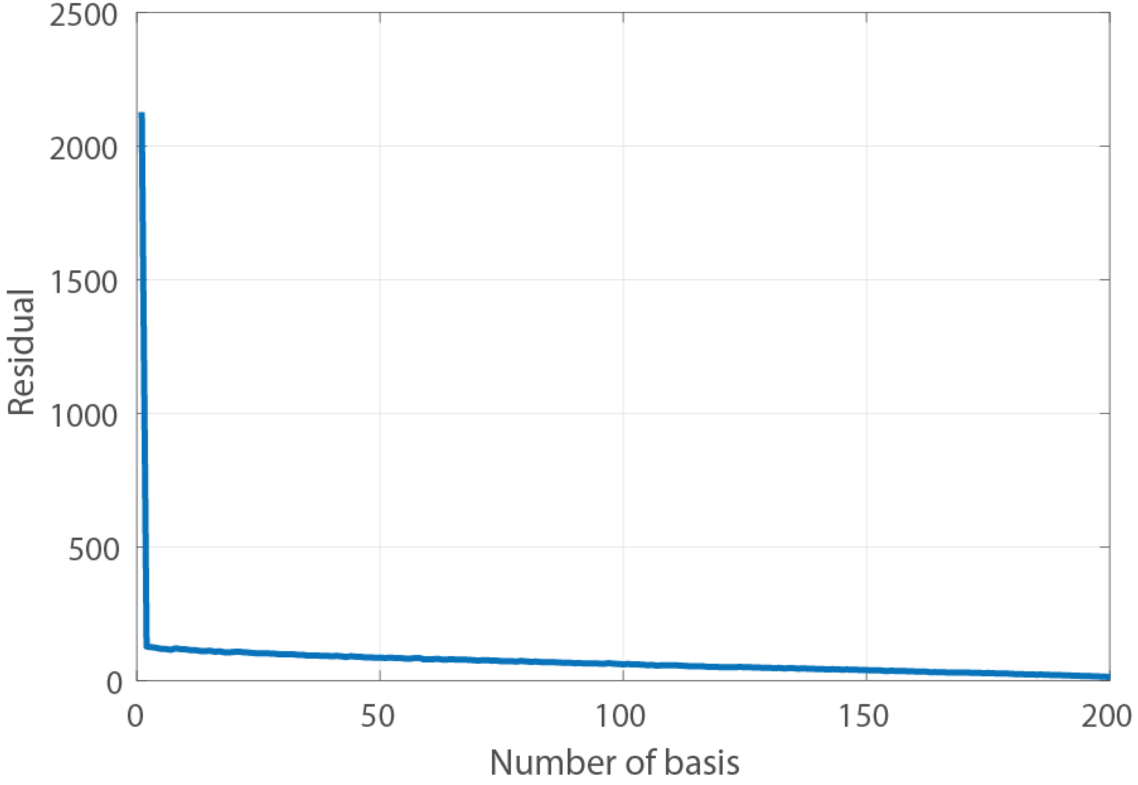
Residual (Nullspace Approximation)



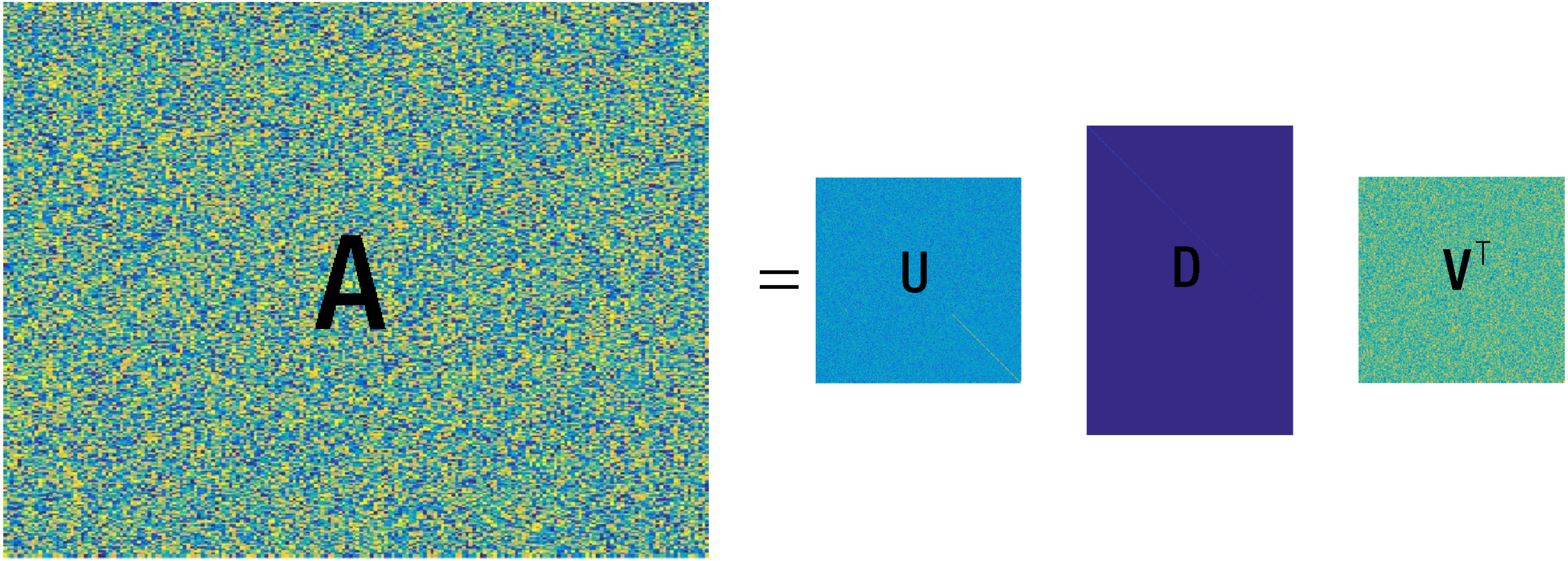
Random matrix

Approximated nullspace of **A:**



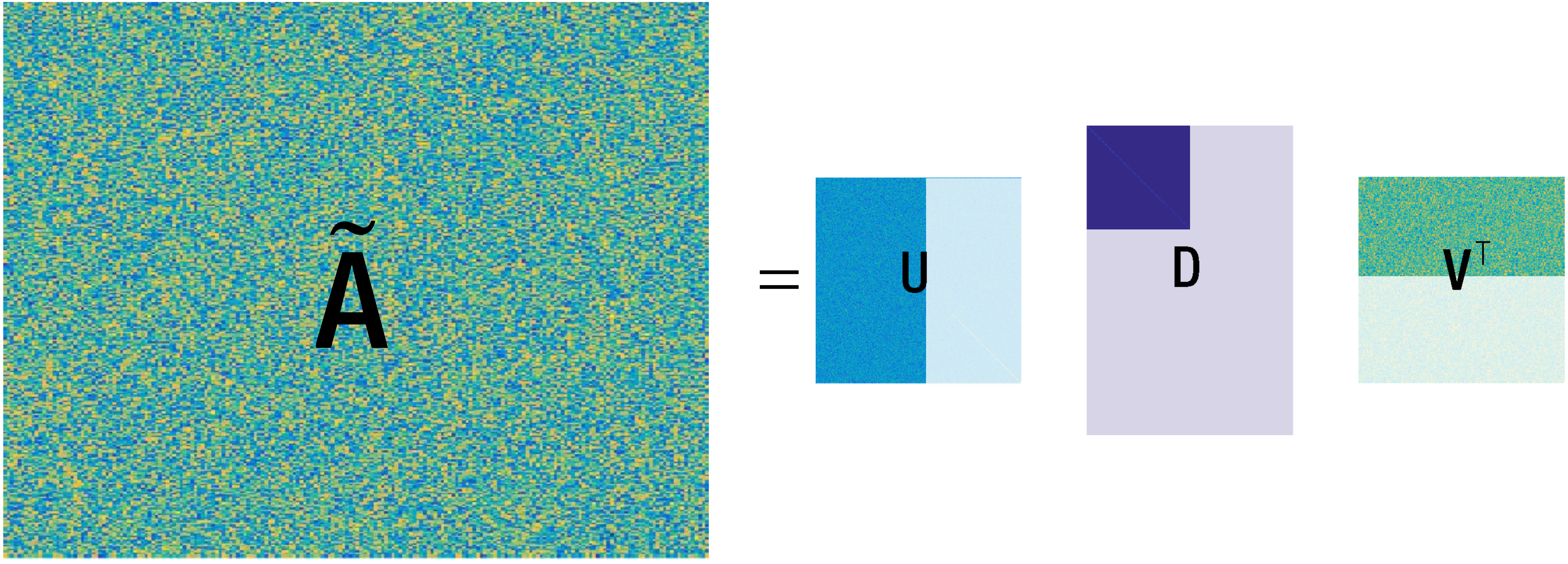


SVD Matrix Approximation



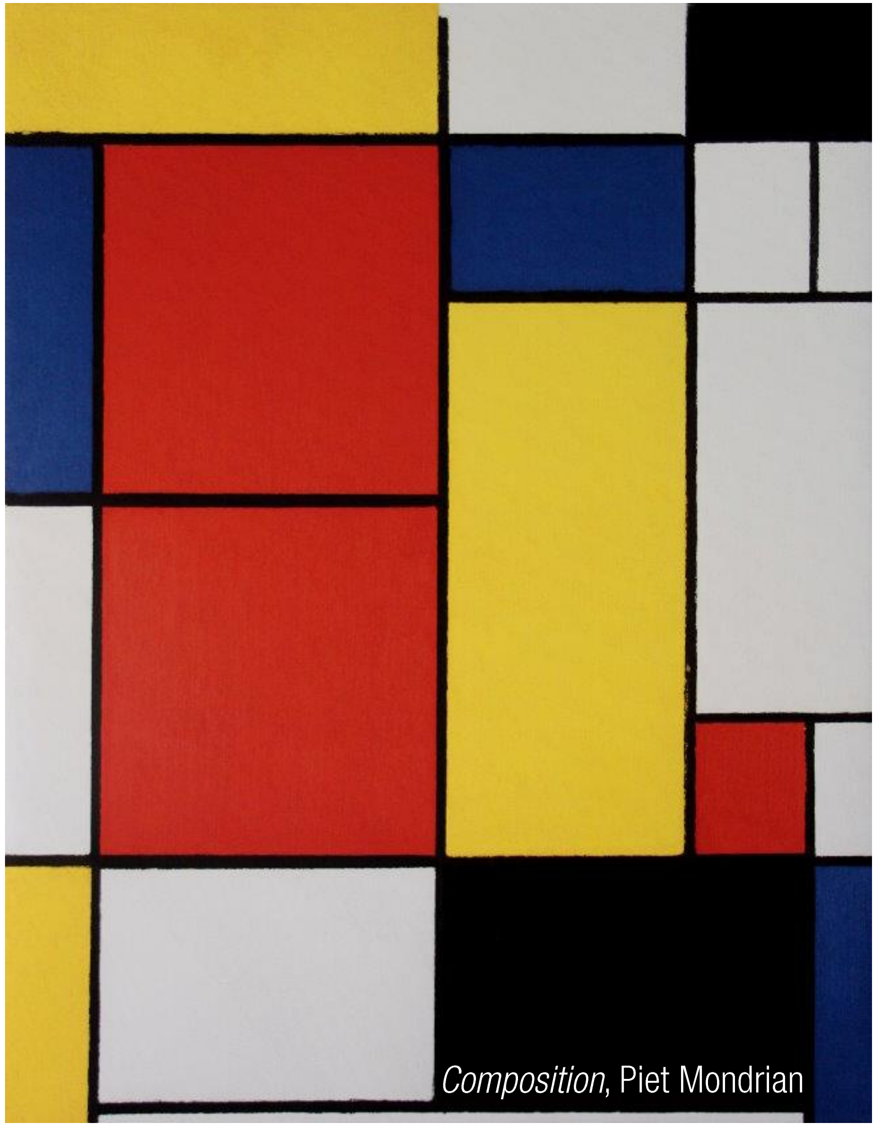
Random matrix

SVD Matrix Approximation

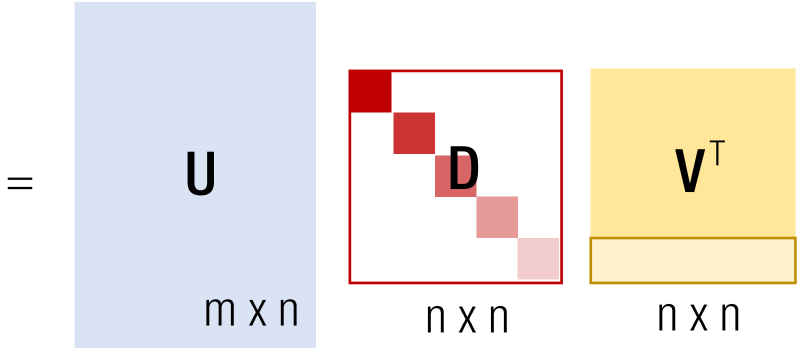


Reconstructed matrix

Mondrian Painting SVD

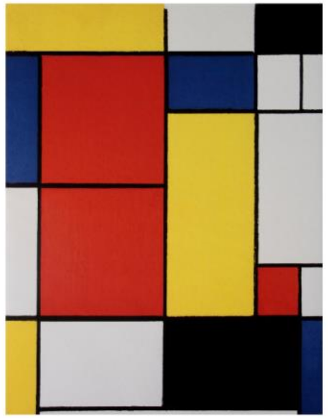


Composition, Piet Mondrian

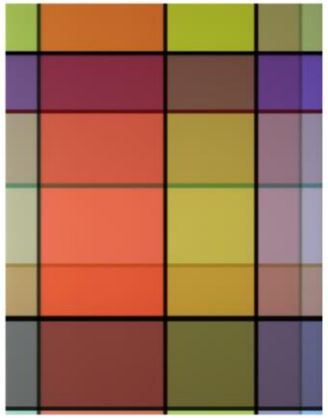


Mondrian Painting SVD Approximation

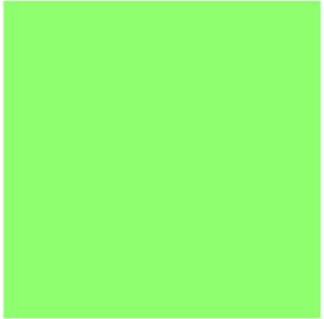
MondrianSVD.m



Ground truth



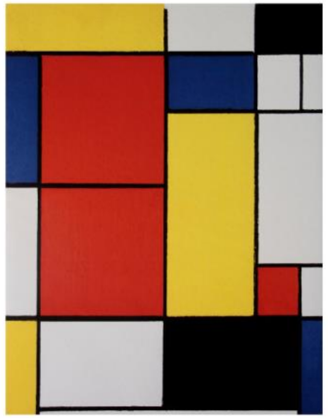
Number of basis: 1



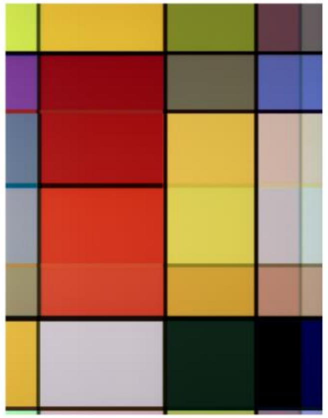
$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

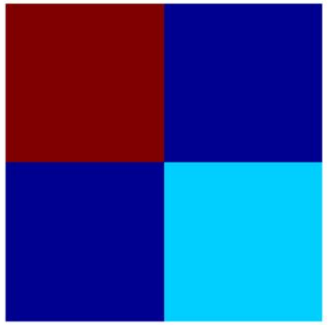
MondrianSVD.m



Ground truth



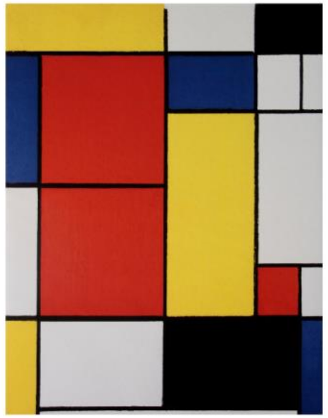
Number of basis: 2



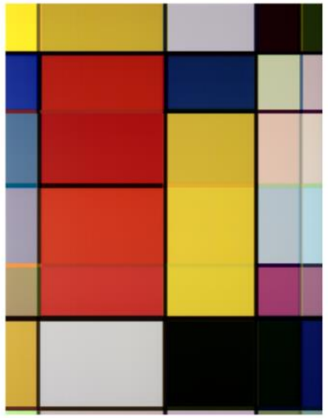
$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

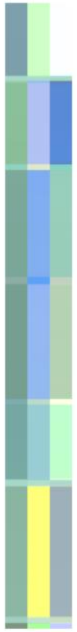
MondrianSVD.m

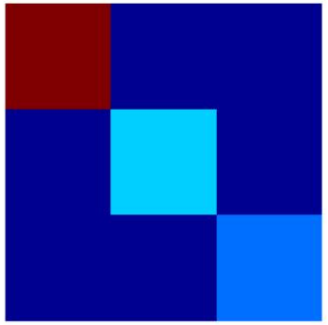


Ground truth



Number of basis: 3

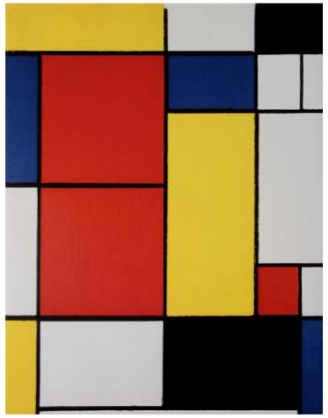




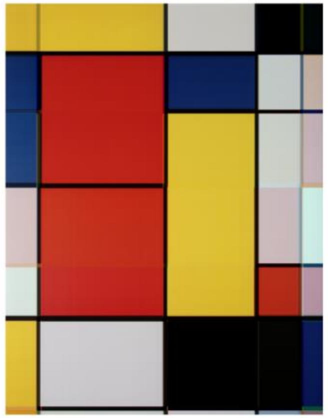
$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

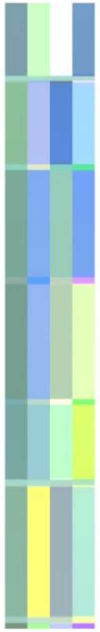
MondrianSVD.m

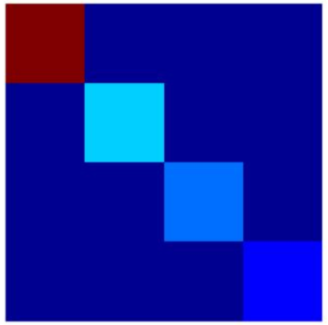


Ground truth



Number of basis: 4

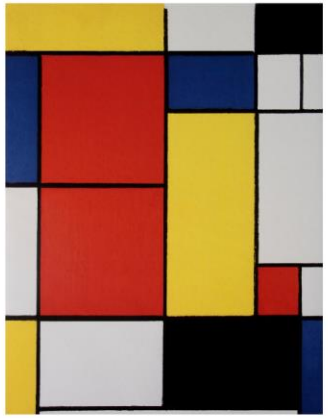




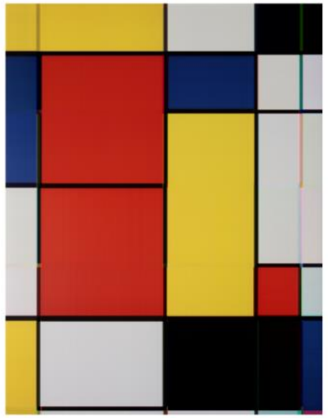
$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

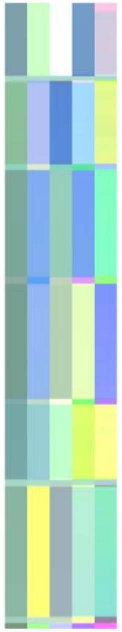
MondrianSVD.m

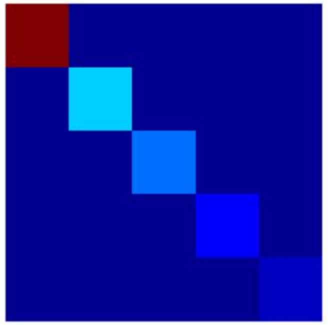


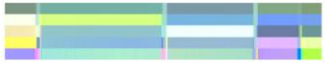
Ground truth



Number of basis: 5



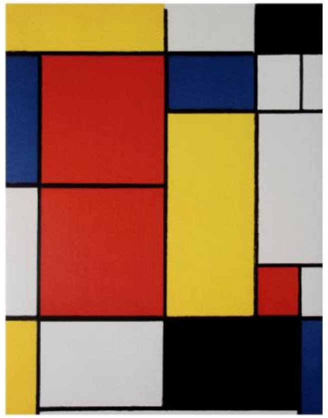




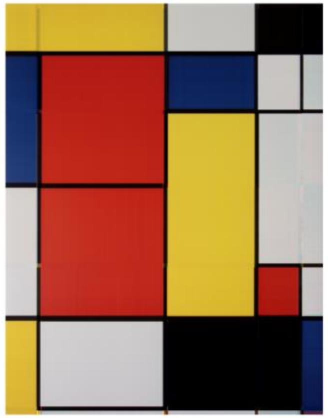
$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

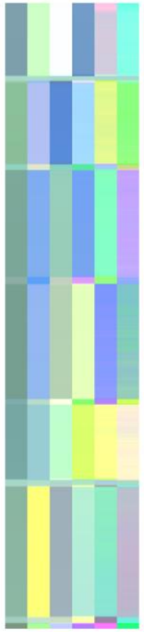
MondrianSVD.m

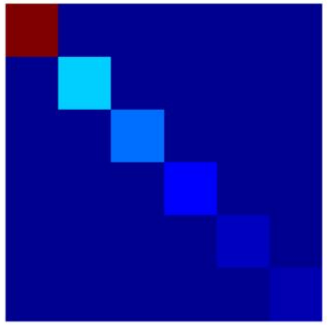


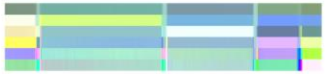
Ground truth



Number of basis: 6



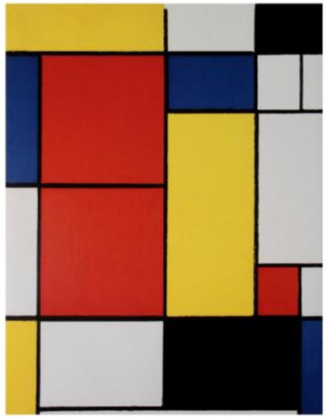




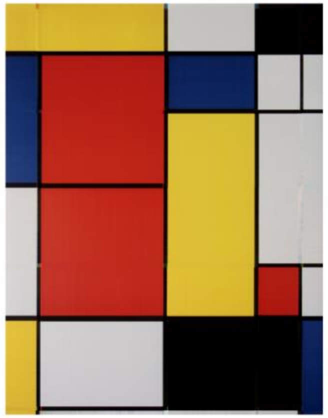
$$A = U \quad D \quad V^T$$

Mondrian Painting SVD Approximation

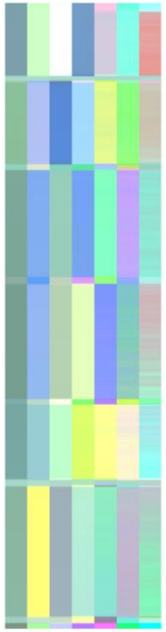
MondrianSVD.m

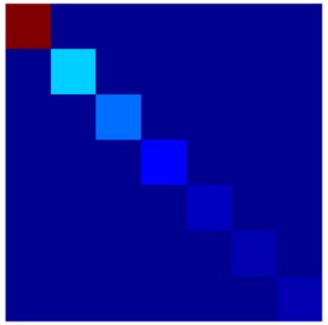


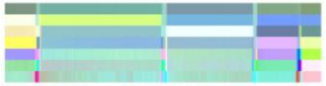
Ground truth



Number of basis: 7



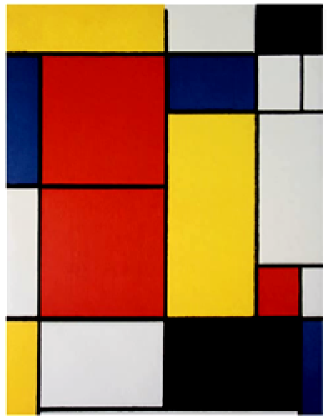




$$A = U \quad D \quad V^T$$

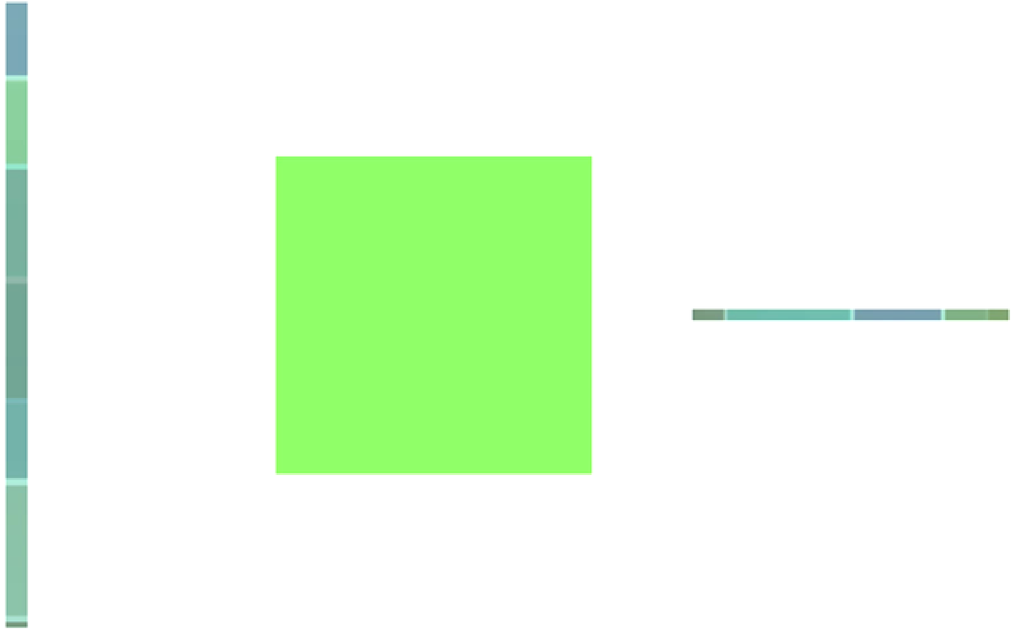
Mondrian Painting SVD Approximation

MondrianSVD.m



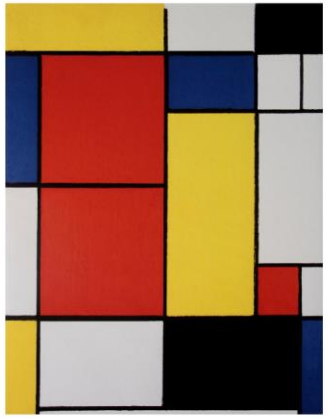
Ground truth

Number of basis: 1

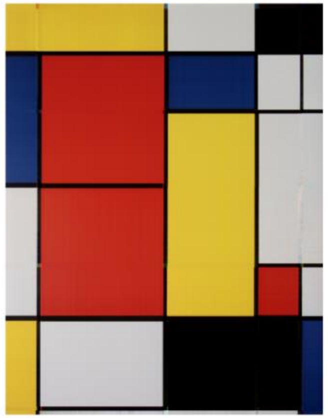


$$A = U \quad D \quad V^T$$

Reconstruction Error



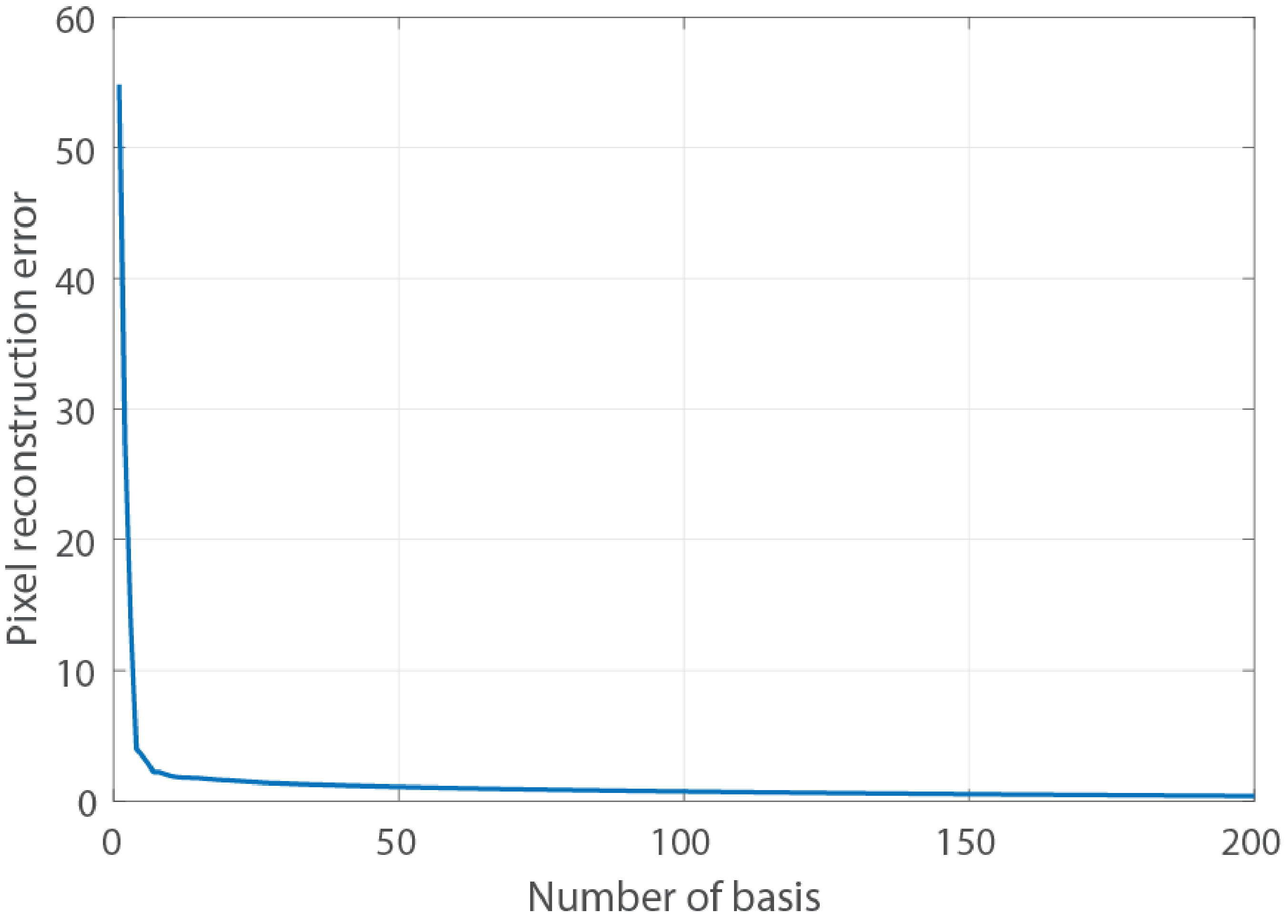
Ground truth



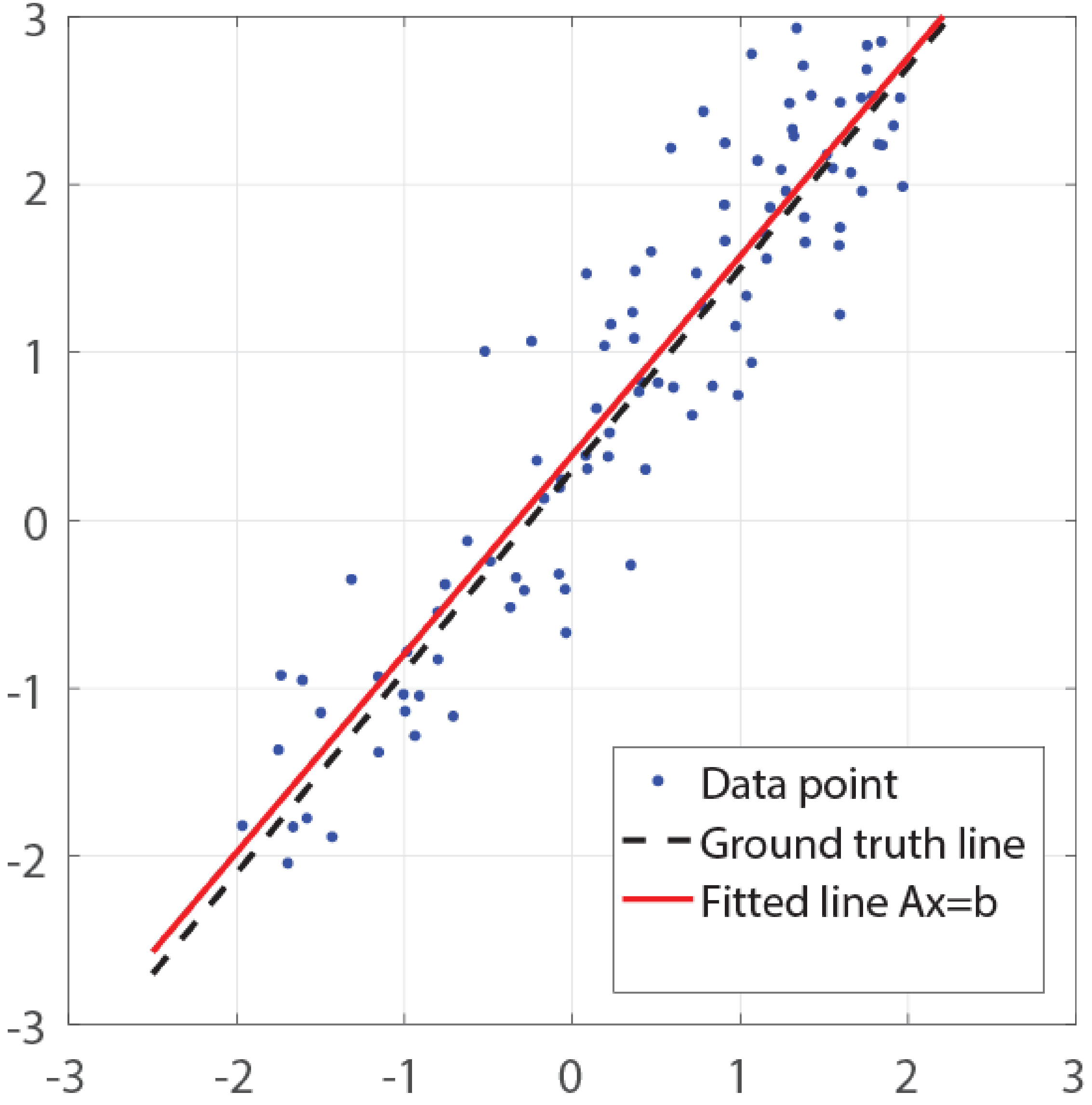
Number of basis: 7

A

MondrianSVD.m



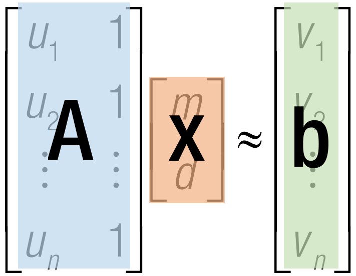
Line Fitting (Ax=b)



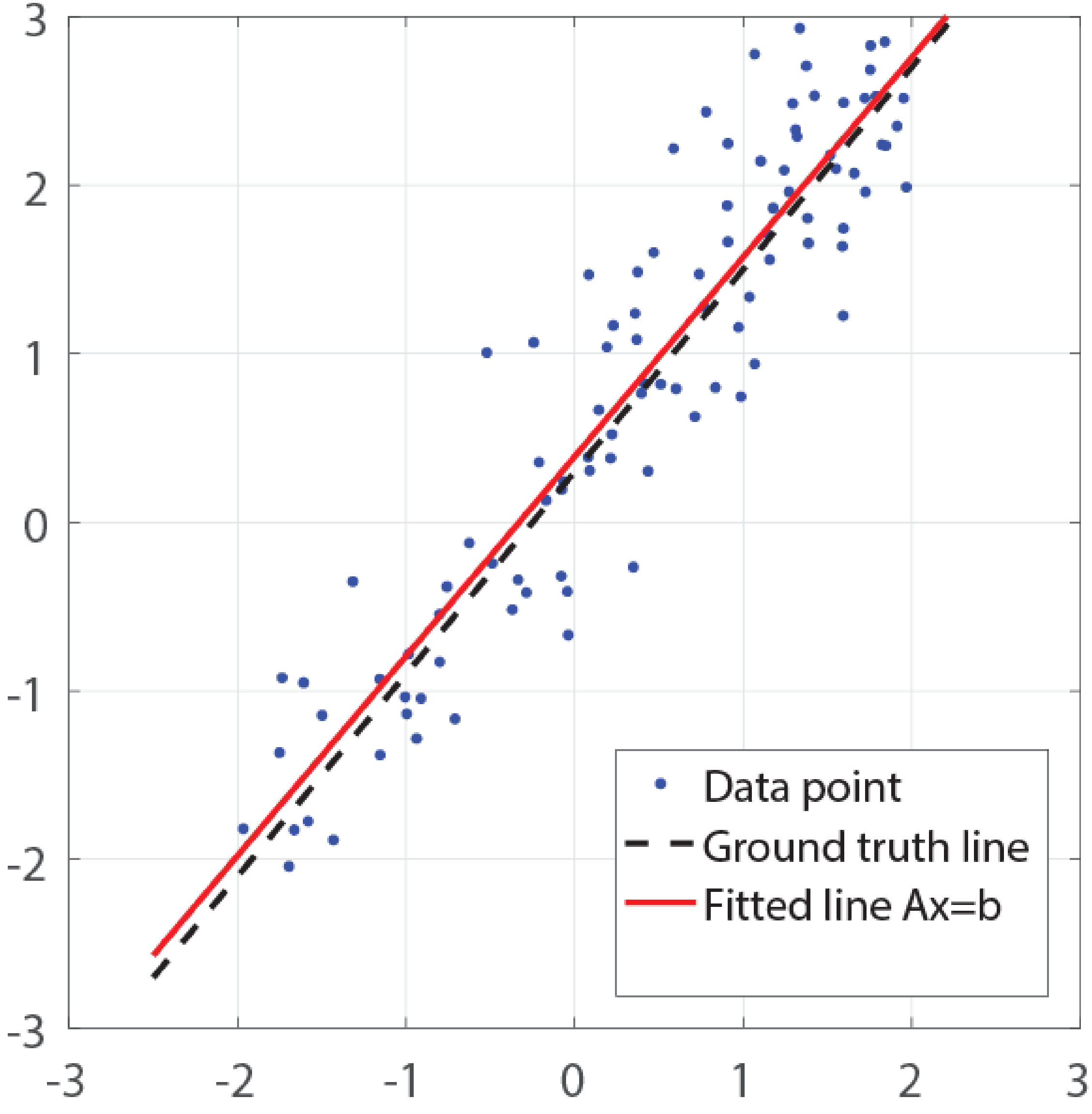
Given points: $(u_1,v_1),(u_2,v_2),\cdots,(u_n,v_n)$

Find the best line: (m,d)

$$v_1 \approx mu_1 + d$$
$$v_2 \approx mu_2 + d$$
$$\vdots$$
$$v_n \approx mu_n + d$$



Line Fitting (Ax=b)



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

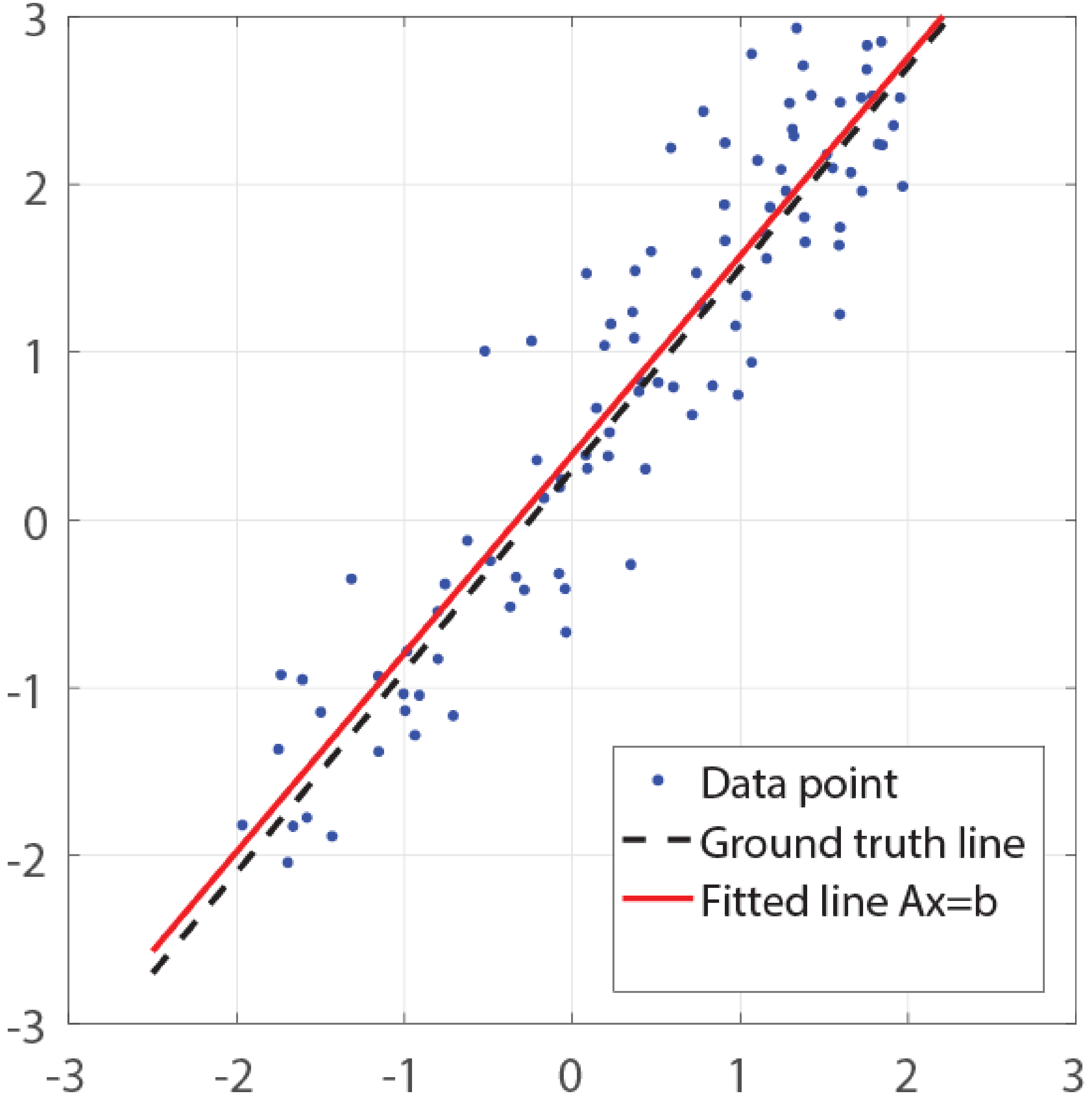
Find the best line: (a, b, c) (m, d)

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

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

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

Line Fitting (Ax=b)



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

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

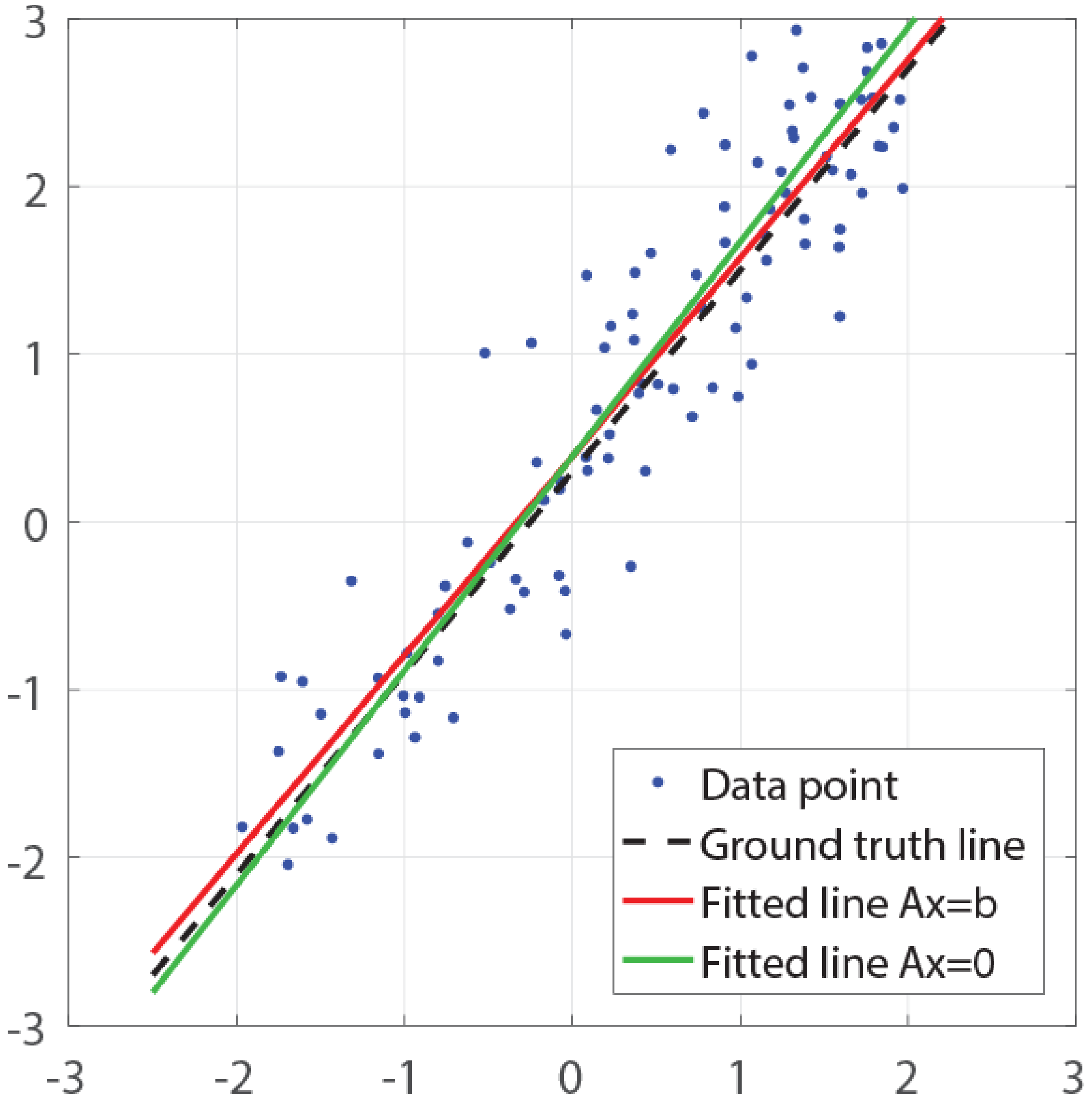
$$a u_1 + b v_1 + c \approx 0$$
$$a u_2 + b v_2 + c \approx 0$$
$$\vdots$$
$$a u_n + b v_n + c \approx 0$$

$$v_1 \approx m u_1 + d$$
$$v_2 \approx m u_2 + d$$
$$\vdots$$
$$v_n \approx m u_n + d$$

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

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

Line Fitting (Ax=b)



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

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

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

$$\underbrace{\begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ \vdots & \vdots & 1 \\ u_n & v_n & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\mathbf{x}} \approx \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{0}} \qquad \underbrace{\begin{bmatrix} u_1 & 1 \\ u_2 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} m \\ d \end{bmatrix}}_{\mathbf{x}} \approx \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{\mathbf{b}}$$

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

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

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

$$\rightarrow \quad h_{11}u_x + h_{12}u_y + h_{13} + h_{31}u_x v_x + h_{32}u_y v_x + h_{33}v_x = 0$$

$$h_{21}u_x + h_{22}u_y + h_{23} + h_{31}u_x v_y + h_{32}u_y v_y + h_{33}v_y = 0$$

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

Unknowns: $h_{11}, \cdots, h_{33}$

Equations: 2 per correspondence

Homography Computation

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

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

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

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

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

Homography Computation

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

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

$$\rightarrow \quad h_{11}u_x + h_{12}u_y + h_{13} - h_{31}u_x v_x - h_{32}u_y v_x - h_{33}v_x = 0$$

$$h_{21}u_x + h_{22}u_y + h_{23} - h_{31}u_x v_y - h_{32}u_y v_y - h_{33}v_y = 0$$

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

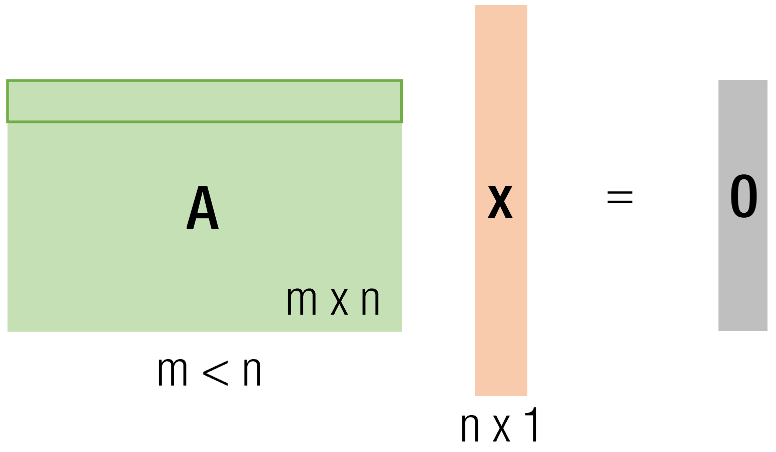
$$\rightarrow \quad \underbrace{\begin{bmatrix} u_x & u_y & 1 & & & & -u_x v_x & -u_y v_x & -v_x \\ & & & u_x & u_y & 1 & -u_x v_y & -u_y v_y & -v_y \end{bmatrix}}_{\mathbf{A}} \; \mathbf{x} = \mathbf{0}, \quad \mathbf{x} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}, \quad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

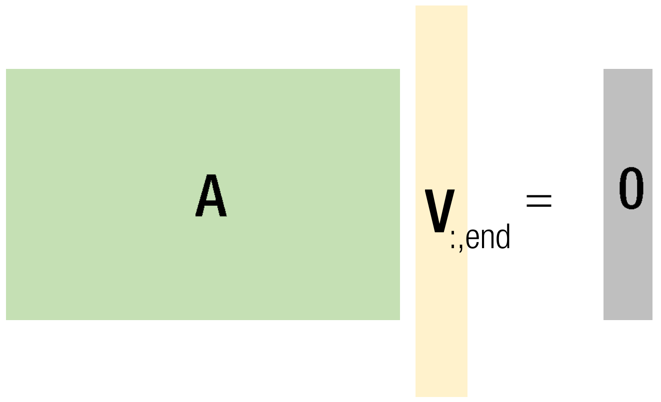
2x9

Recall: Singular Value Decomposition (SVD)

eqs < # unknowns

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





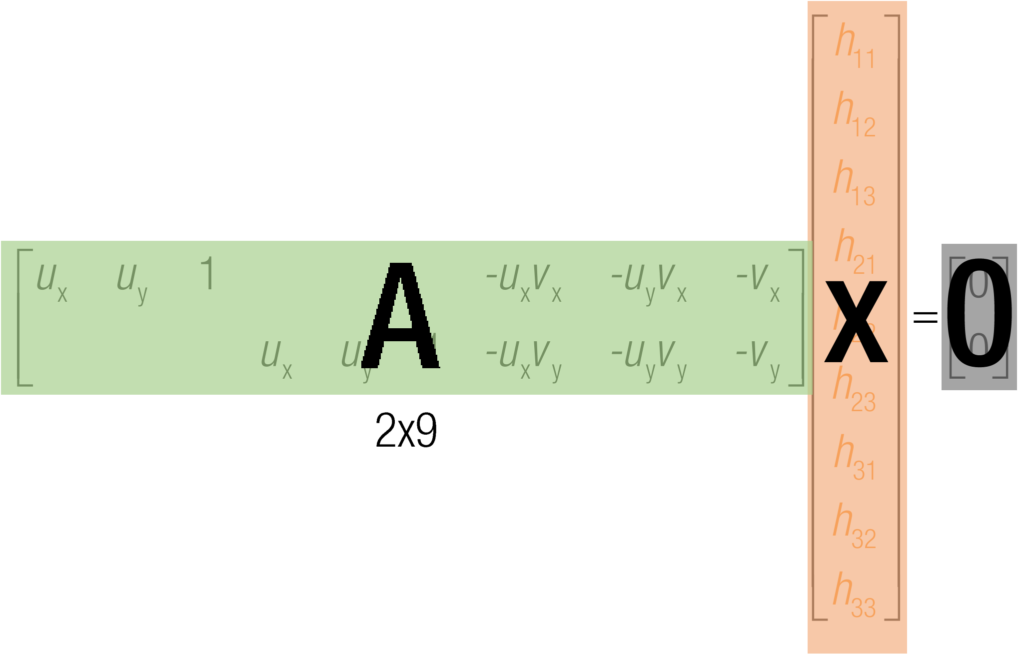
$$\mathbf{V}_{:end} = \text{null}(\mathbf{A})$$

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

For a unique solution, **A** should be m x (m+1)

Homography Computation

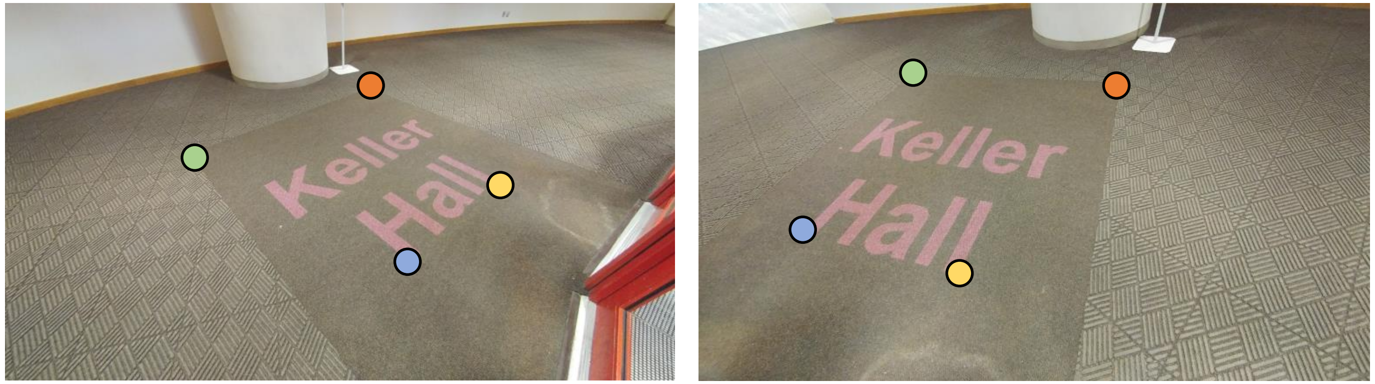
How many correspondences are needed?



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

Homography Computation

How many correspondences are needed? 4



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

$$\underbrace{\begin{bmatrix} u_x & u_y & 1 & & & & -u_x v_x & -u_y v_x & -v_x \\ & & & u_x & u_y & 1 & -u_x v_y & -u_y v_y & -v_y \end{bmatrix}}_{\mathbf{A}\ (8\text{x}9)} \underbrace{\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}}_{\mathbf{x}} = \mathbf{0}$$

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

Homography Computation

How many correspondences are needed? 4

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

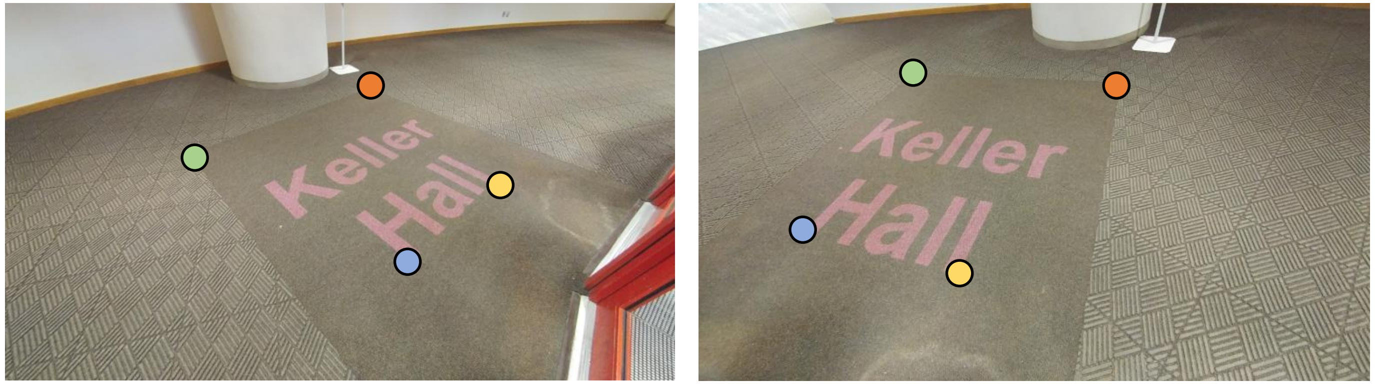
ComputeHomography.m

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

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