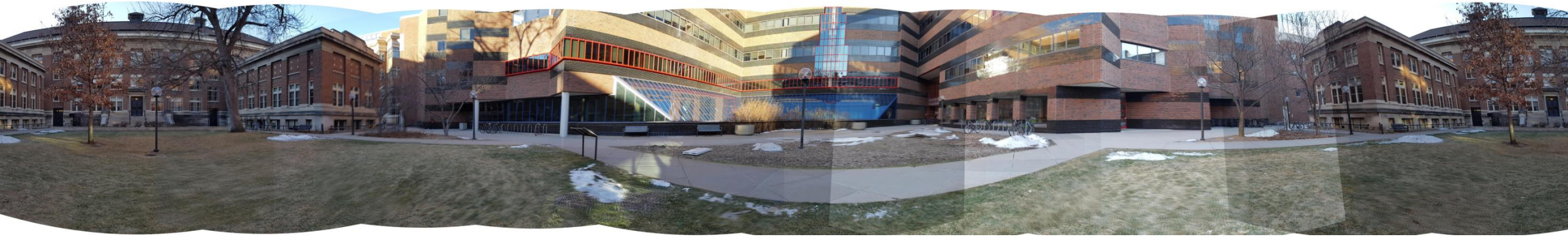
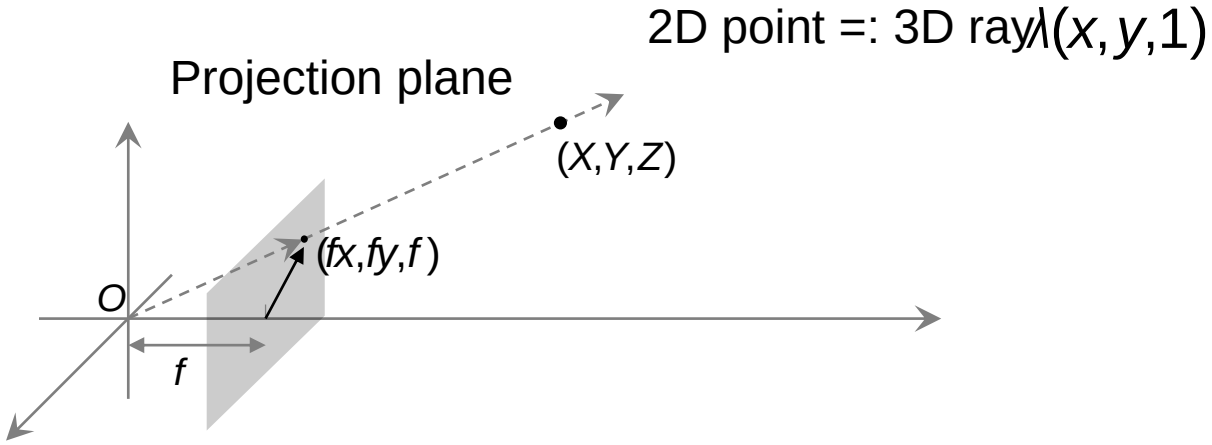
Video 5.1
Jianbo Shi

Image Transform



Homogeneous Coordinate



$$(x,y) \rightarrow (x,y,1)$$

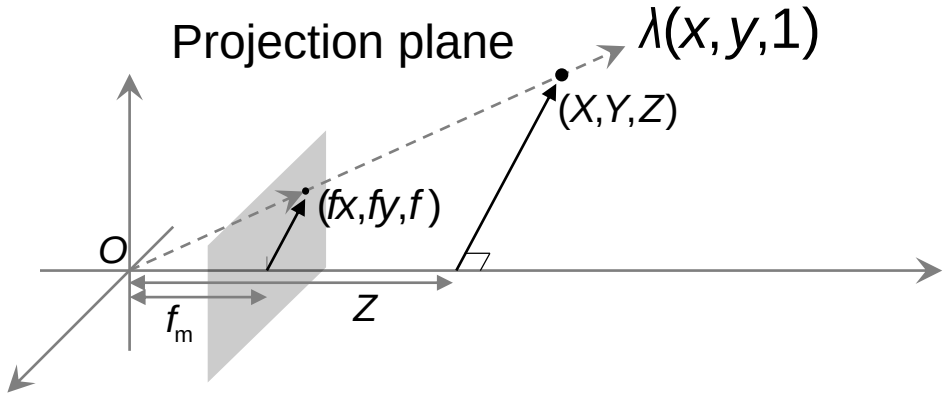
: A point in Euclidean space ($\mathbb{R}^2$) can be represented by a homogeneous representation in Projective space ($\mathbb{P}^2$) (3 numbers).

$$= f(x,y,1)$$

$$= \lambda(x,y,1) = (X,Y,Z)$$

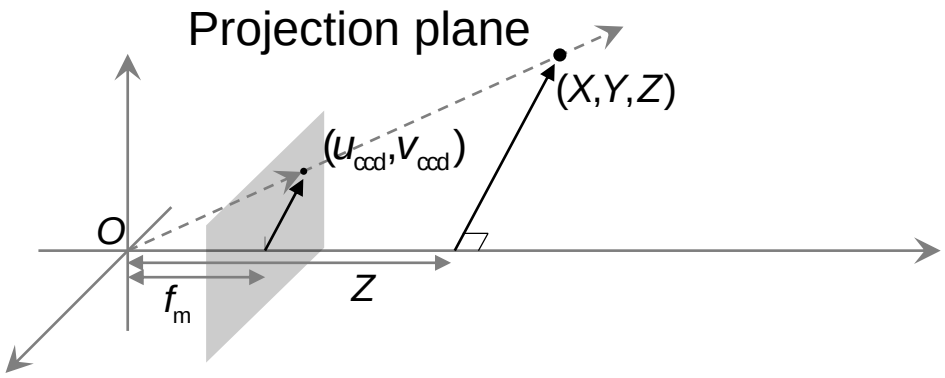
3D Point Projection (Metric Space)

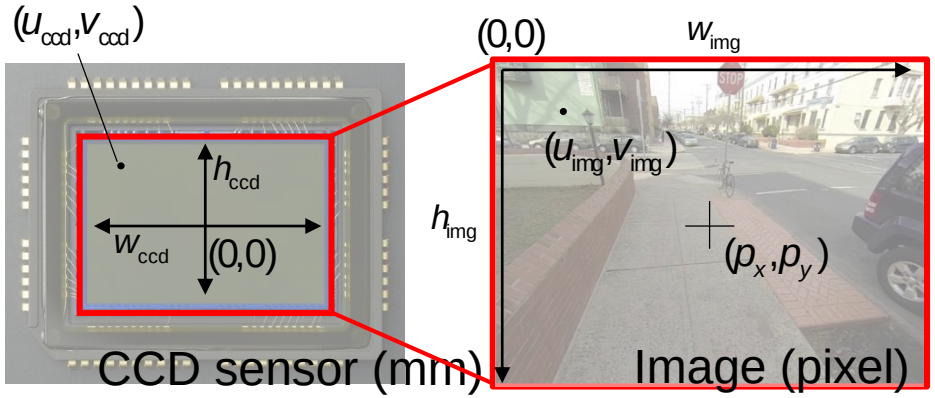
2D point =: 3D ray



$$(x, y, 1) = (f_m x, f_m y, f_m) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z}, f_m)$$

3D Point Projection (Pixel Space)

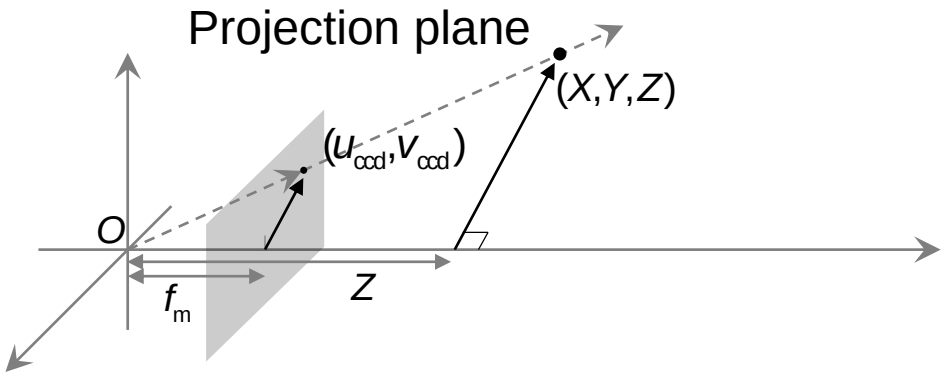




$$(X,Y,Z) \rightarrow (u_{ccd}, v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

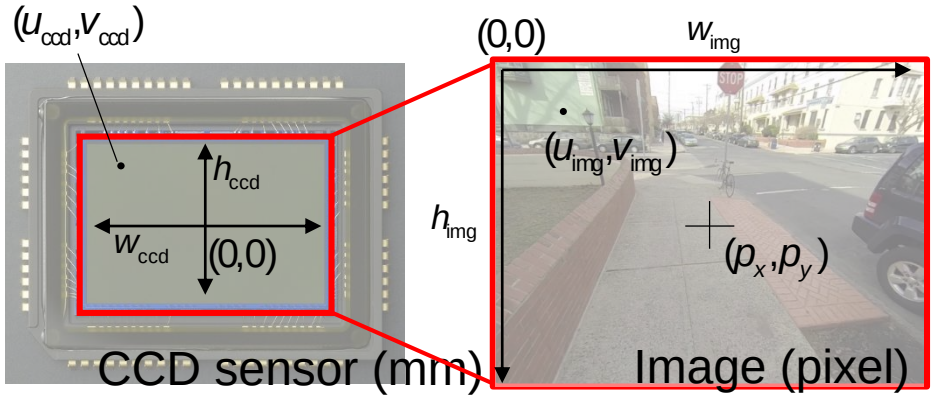
$$u_{img} = f_x \frac{X}{Z} + p_x \qquad v_{img} = f_y \frac{Y}{Z} + p_y$$

3D Point Projection (Pixel Space)



$$(X,Y,Z) \rightarrow (u_{ccd},v_{ccd}) = (f_m \frac{X}{Z}, f_m \frac{Y}{Z})$$

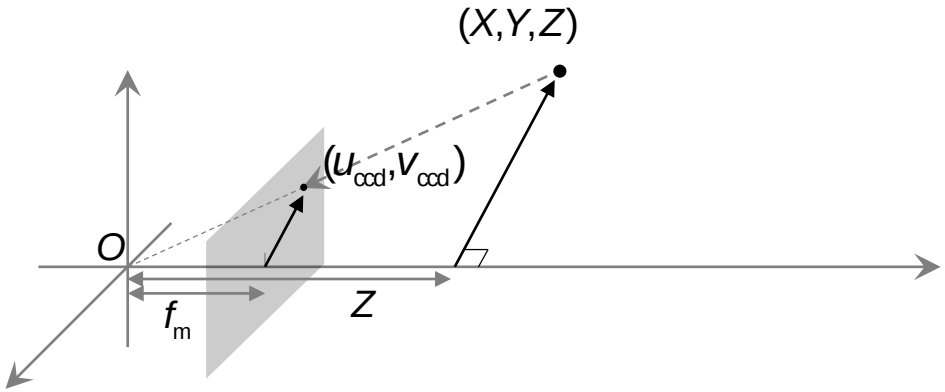
$$u_{img} = f_x \frac{X}{Z} + p_x \qquad v_{img} = f_y \frac{Y}{Z} + p_y$$



$$\lambda \begin{bmatrix} u_{img} \\ v_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Homogeneous representation

Camera Intrinsic Parameter

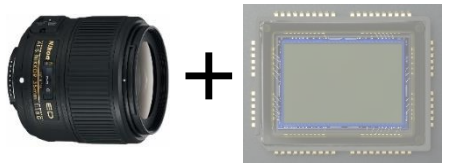


Projection plane

Pixel space Metric space

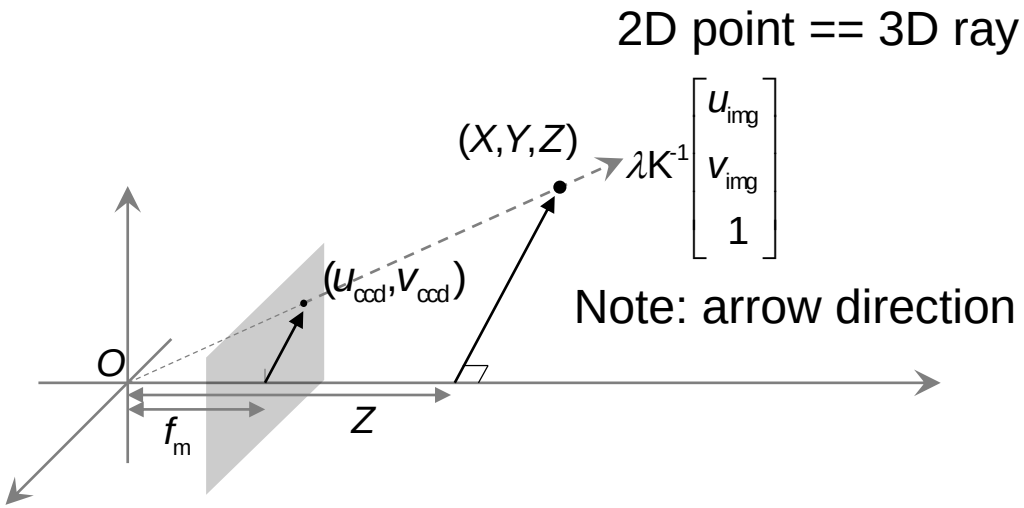
$$\lambda \begin{bmatrix} u_{img} \\ v_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

K



Camera intrinsic parameter
: metric space to pixel space

2D Inverse Projection



Pixel space Metric space

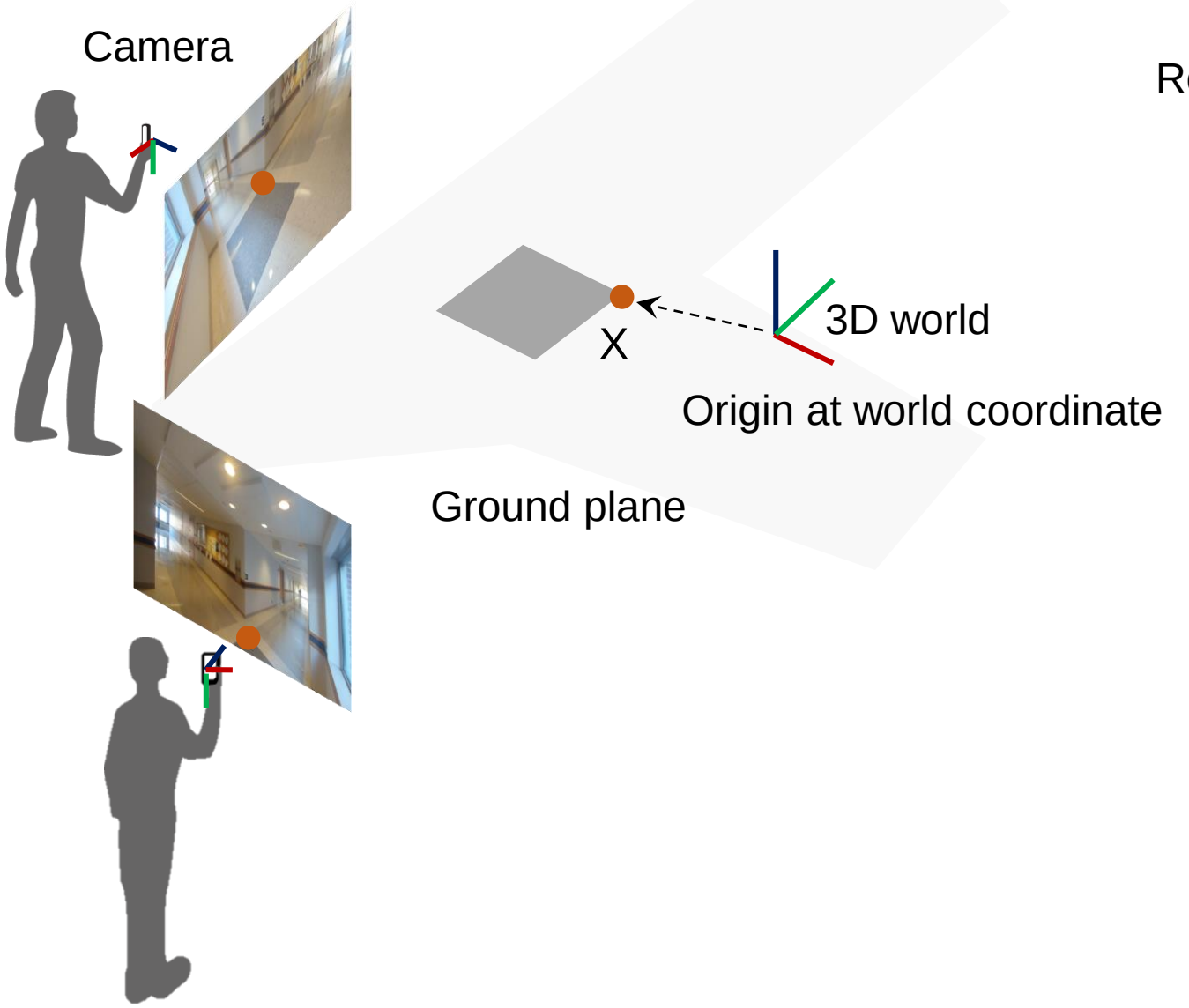
$$\lambda \begin{bmatrix} u_{img} \\ v_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\lambda K^{-1} \begin{bmatrix} u_{img} \\ v_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

3D ray

The 3D point must lie in
the *3D ray* passing through the origin and 2D image point.

Camera Model (3rd Person Perspective)



Recall camera projection matrix:

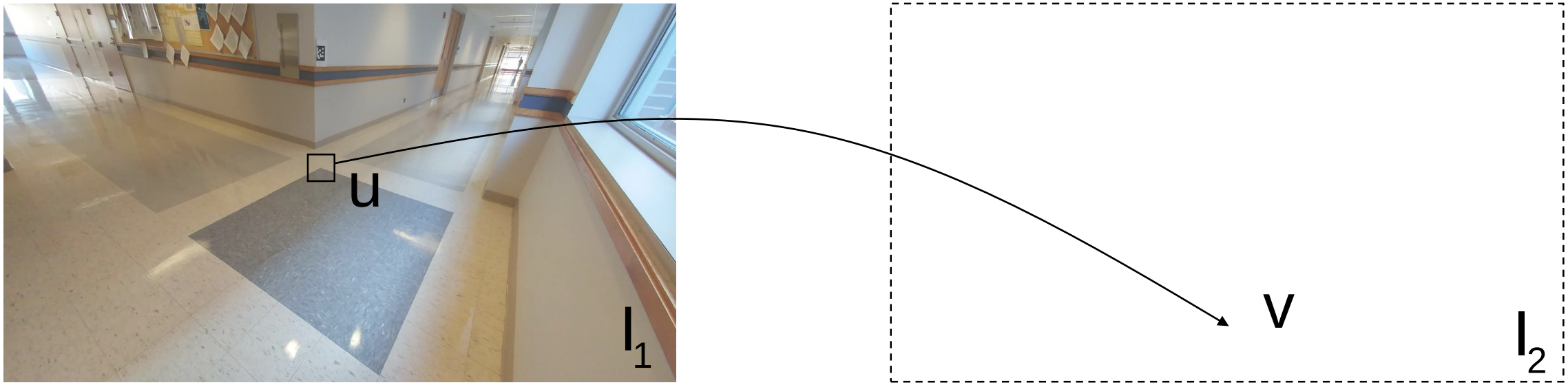
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

K

2D image (pix) 3D world (metric)

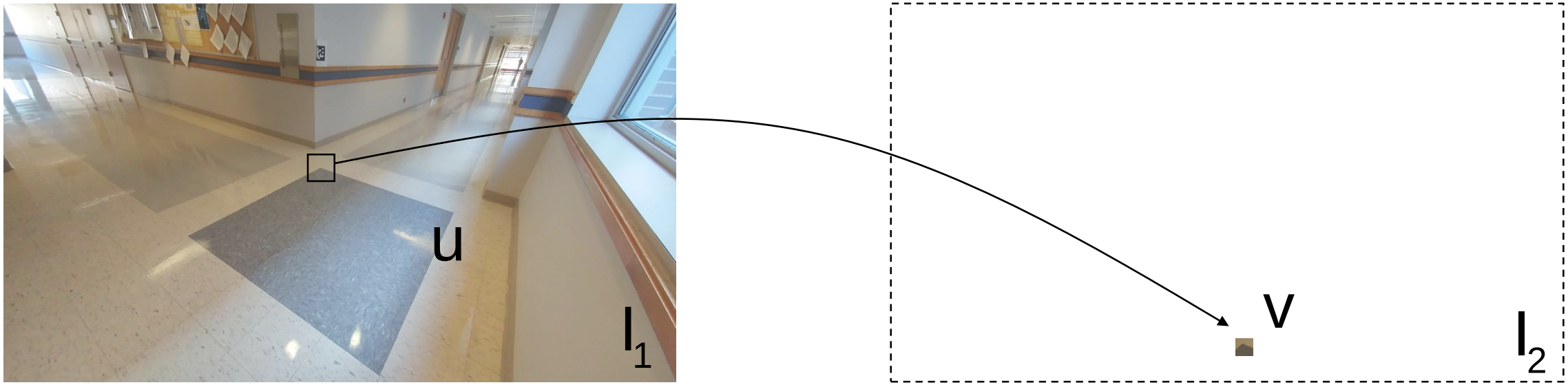
Video 5.2
Jianbo Shi

Image Warping (Coordinate Transform)



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

Image Warping (Coordinate Transform)



$I_2(v) = I_1(u)$: Pixel transport

Uniform Scaling

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

Uniform Scaling

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

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = ? \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Uniform Scaling

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

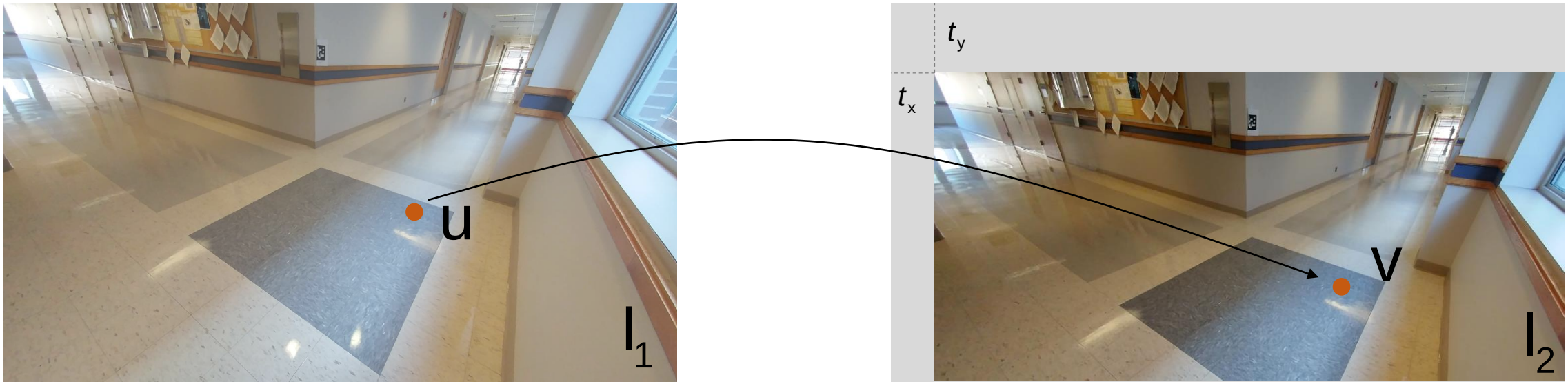
$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & & \\ & s_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} \qquad s_x = s_y$$

Aspect Ratio Change

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

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & & \\ & s_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} \qquad s_x \neq s_y$$

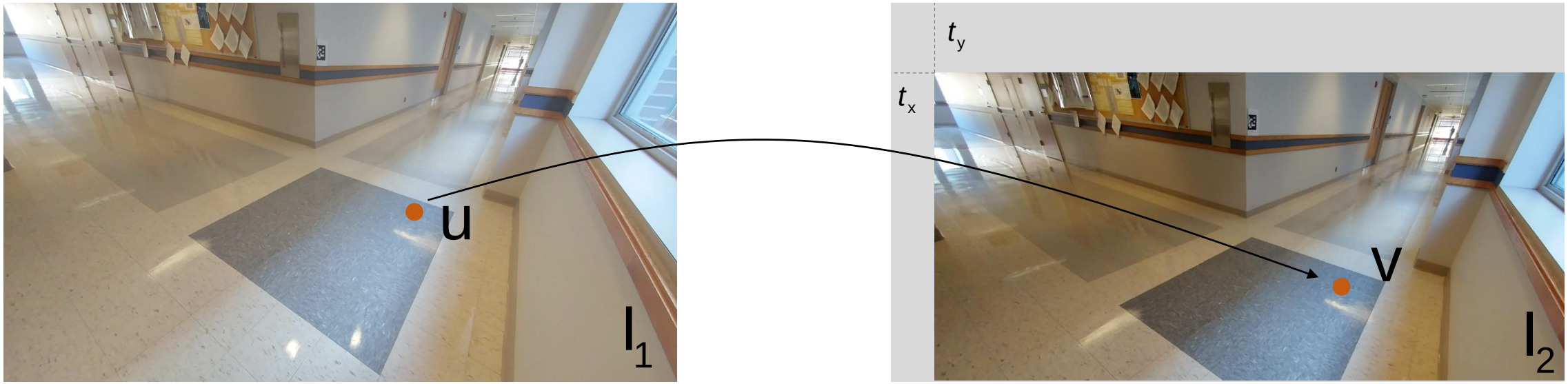
Translation



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

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = ? \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

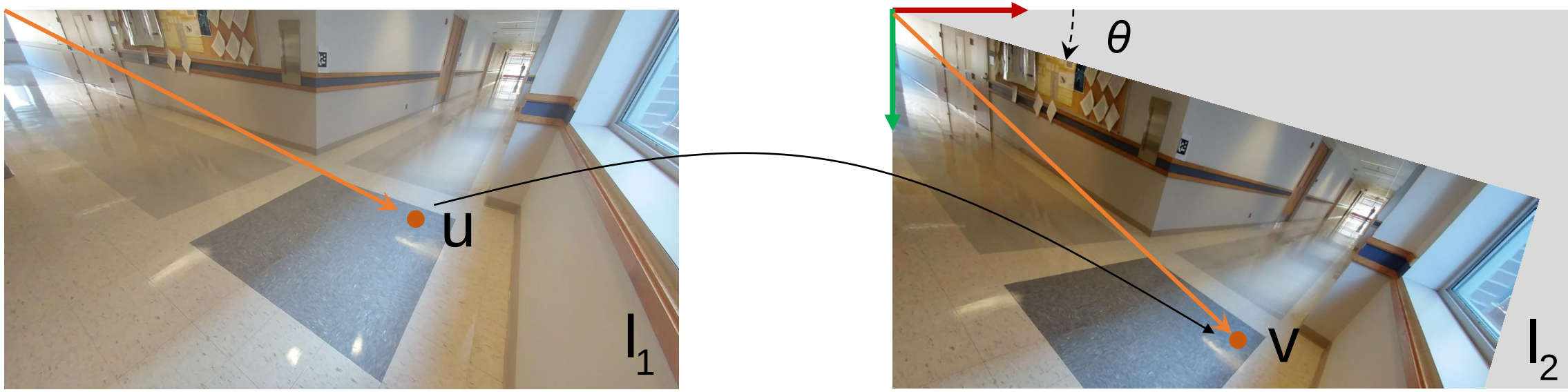
Translation



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

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & & t_x \\ & 1 & t_y \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Rotation

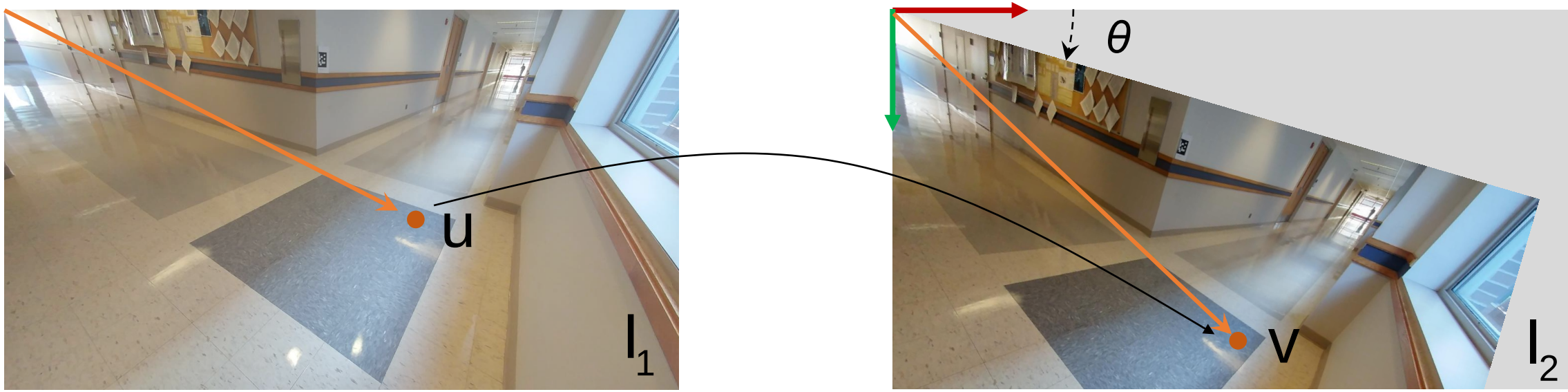


$$I_2(v) = I_1(u)$$

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = ? \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Propert

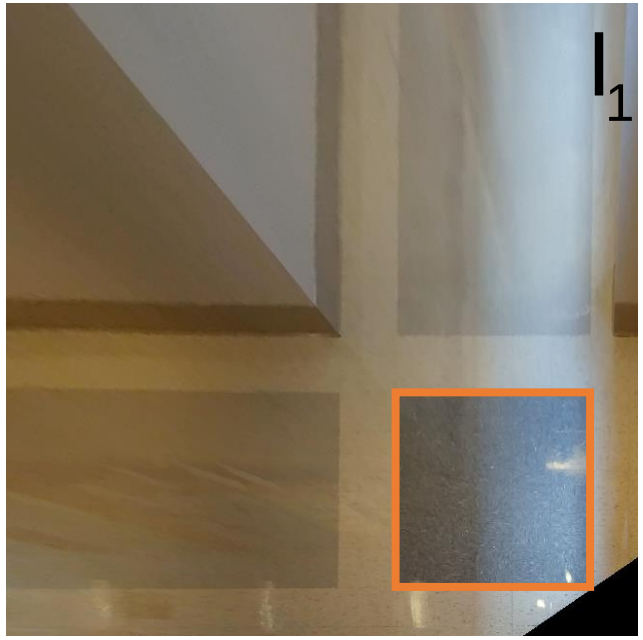
Rotation

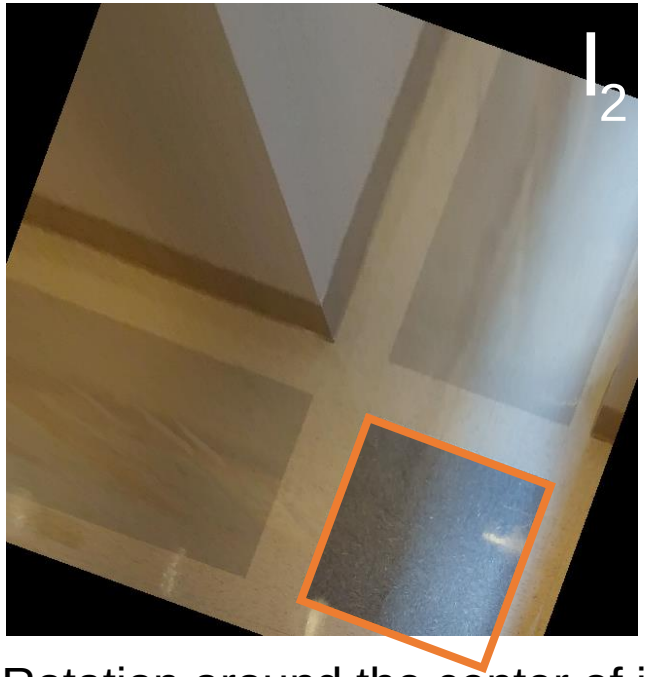


$$I_2(v) = I_1(u)$$

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & \\ \sin\theta & \cos\theta & \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Euclidean Transform SE(2)

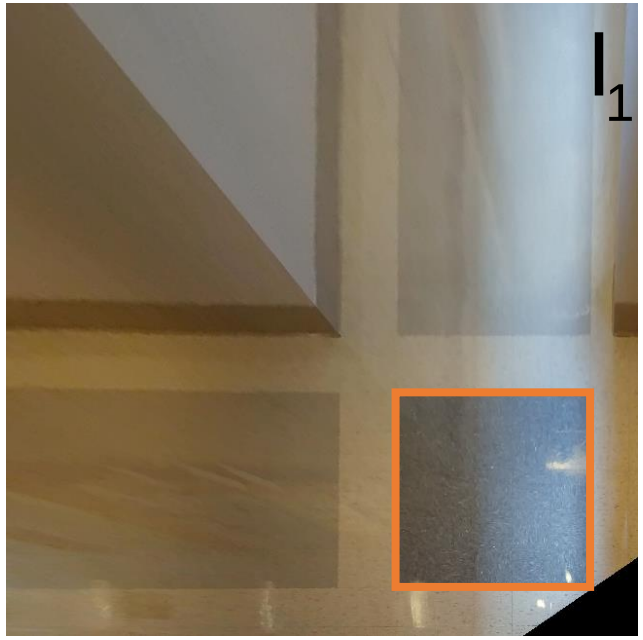


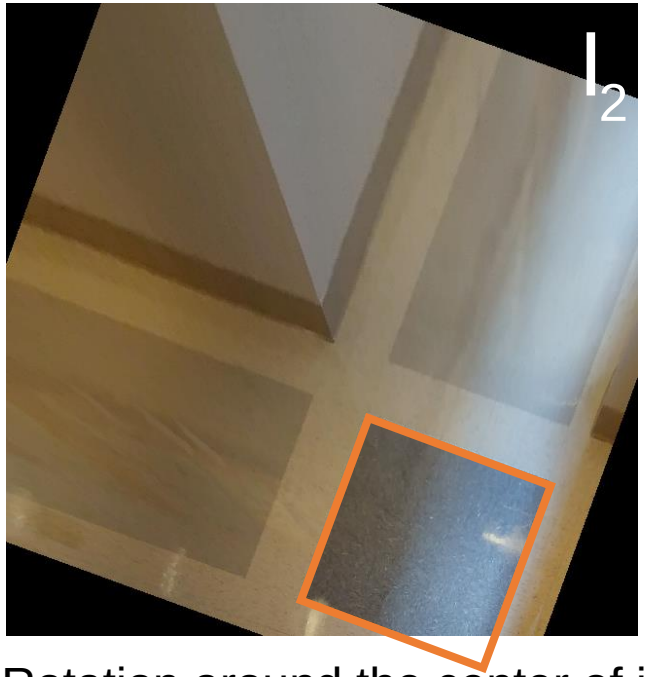


Rotation around the center of image

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \quad ? \quad \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Euclidean Transform SE(2)

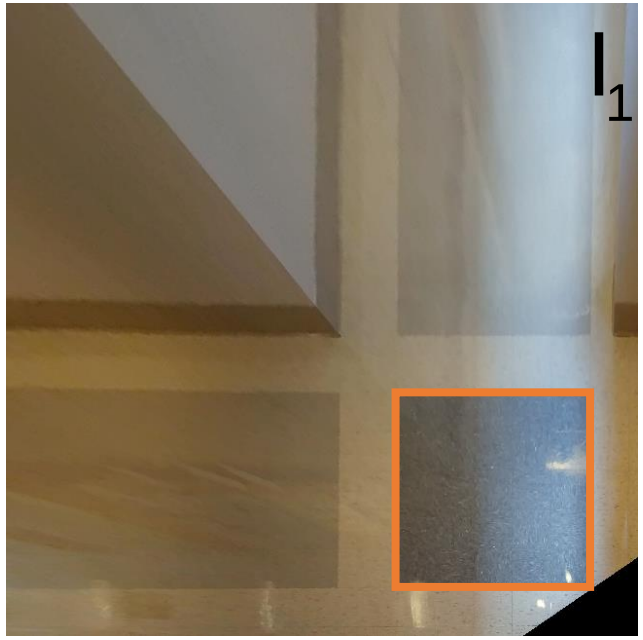


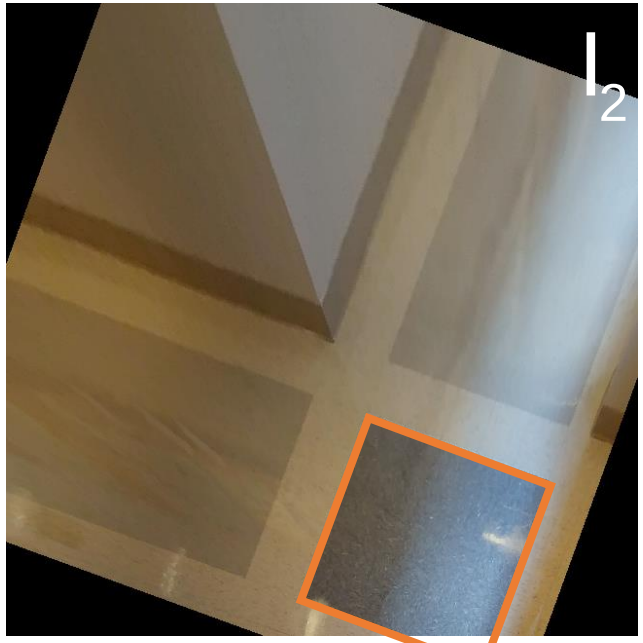


Rotation around the center of image

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Euclidean Transform SE(2)





Rotation around the center of image

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

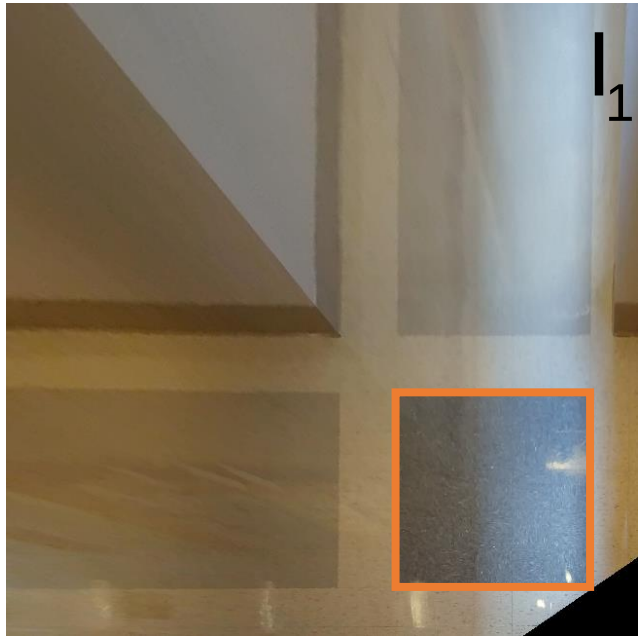
Invariant properties

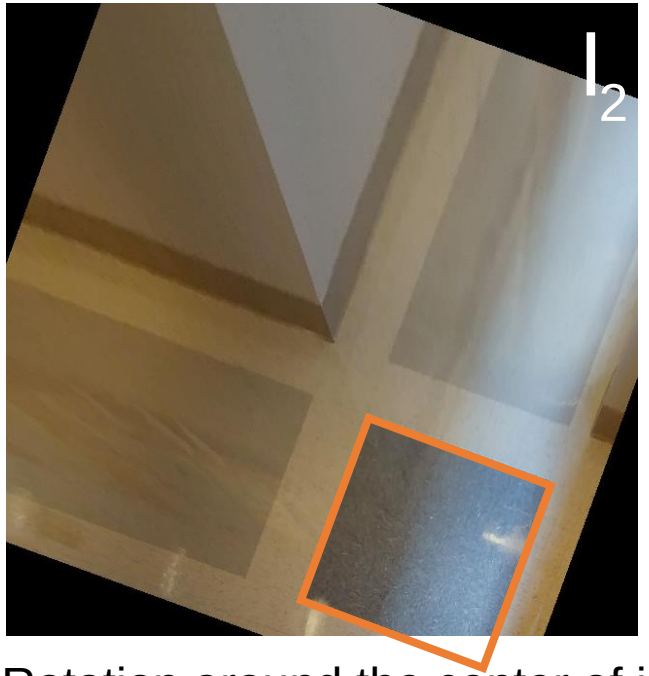
- Length
- Angle
- Area

Degree of freedom

3 (2 translation+1 rotation)

Euclidean Transform SE(2)





Rotation around the center of image

Invariant properties

- Length
- Angle
- Area

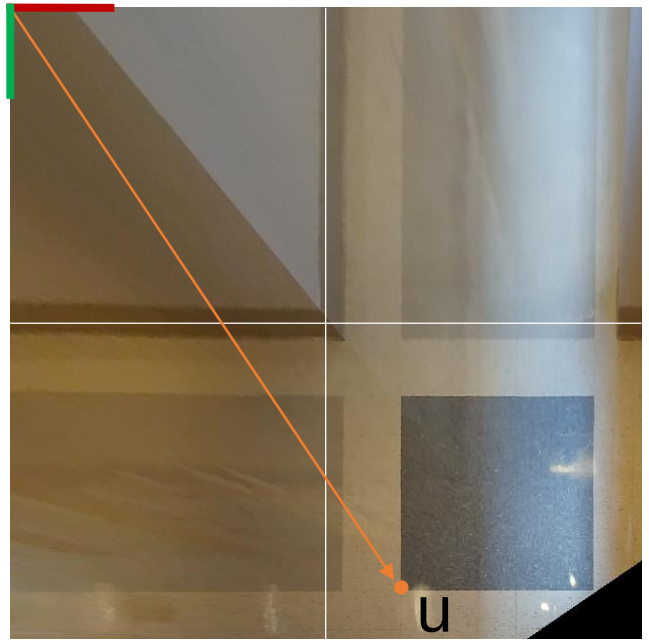
Degree of freedom

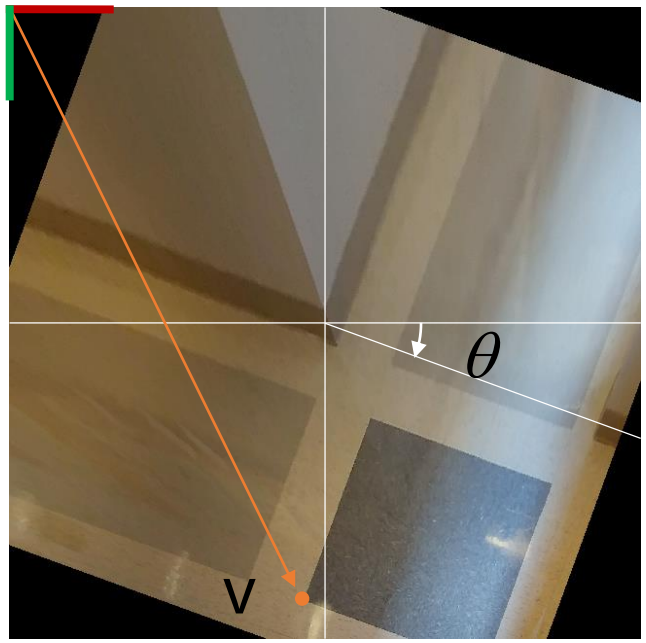
3 (2 translation+1 rotation)

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix}$$

Video 5.3
Jianbo Shi

Euclidean Transform SE(2)



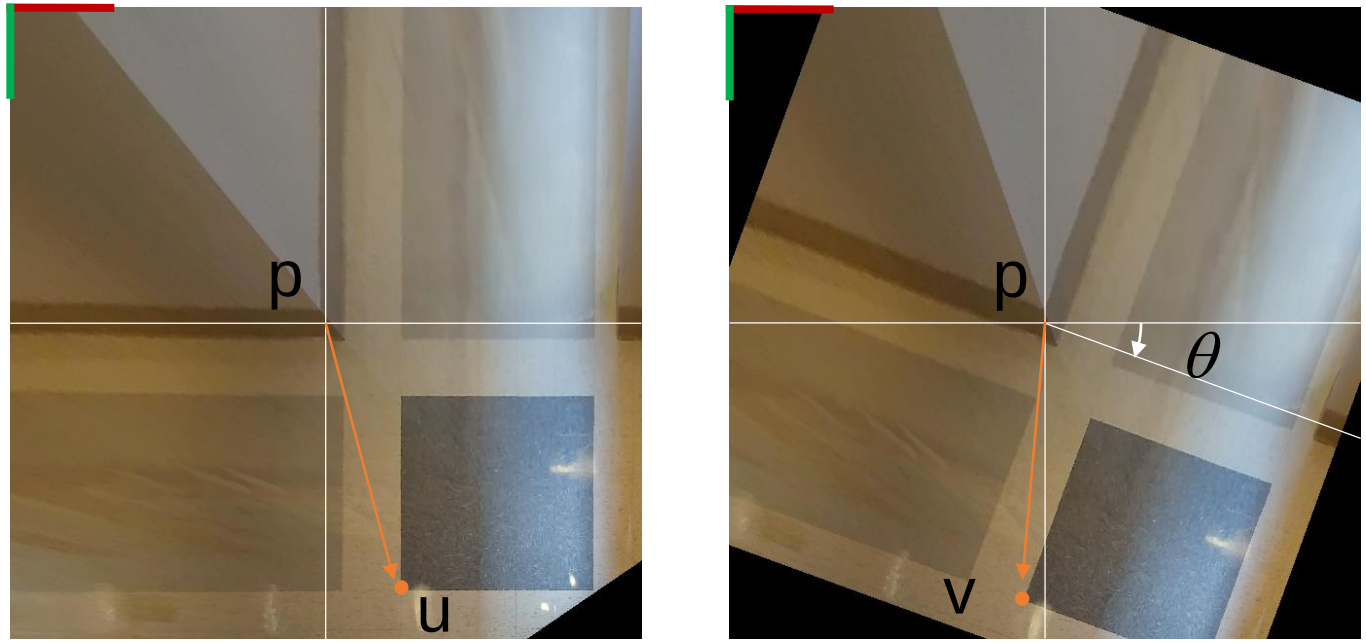


Rotate about the image center

$$\begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \longrightarrow v = Ru + t$$

t ?

Euclidean Transform SE(2)



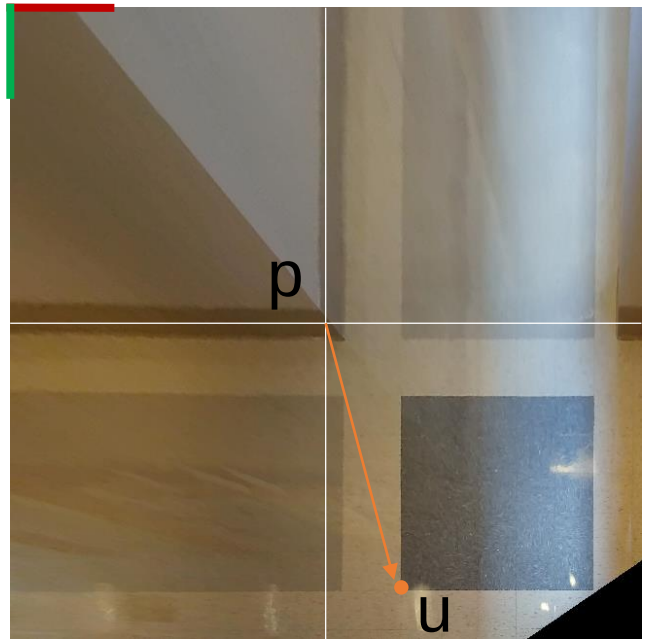
Rotate about the image center

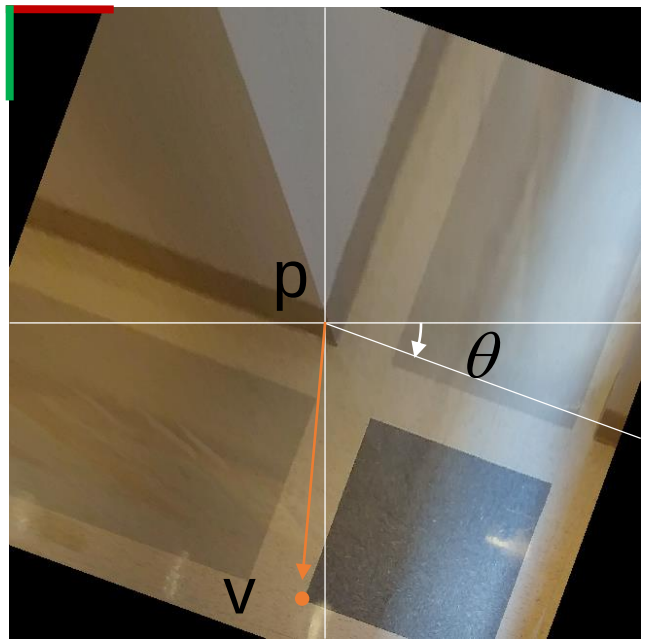
$$\begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \longrightarrow v = Ru + t$$

$$\bar{u} = u - p \qquad \bar{v} = v - p$$

$$\longrightarrow \quad \bar{v} = R\bar{u}$$

Euclidean Transform SE(2)





Rotate about the image center

$$\begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \longrightarrow v = Ru + t$$

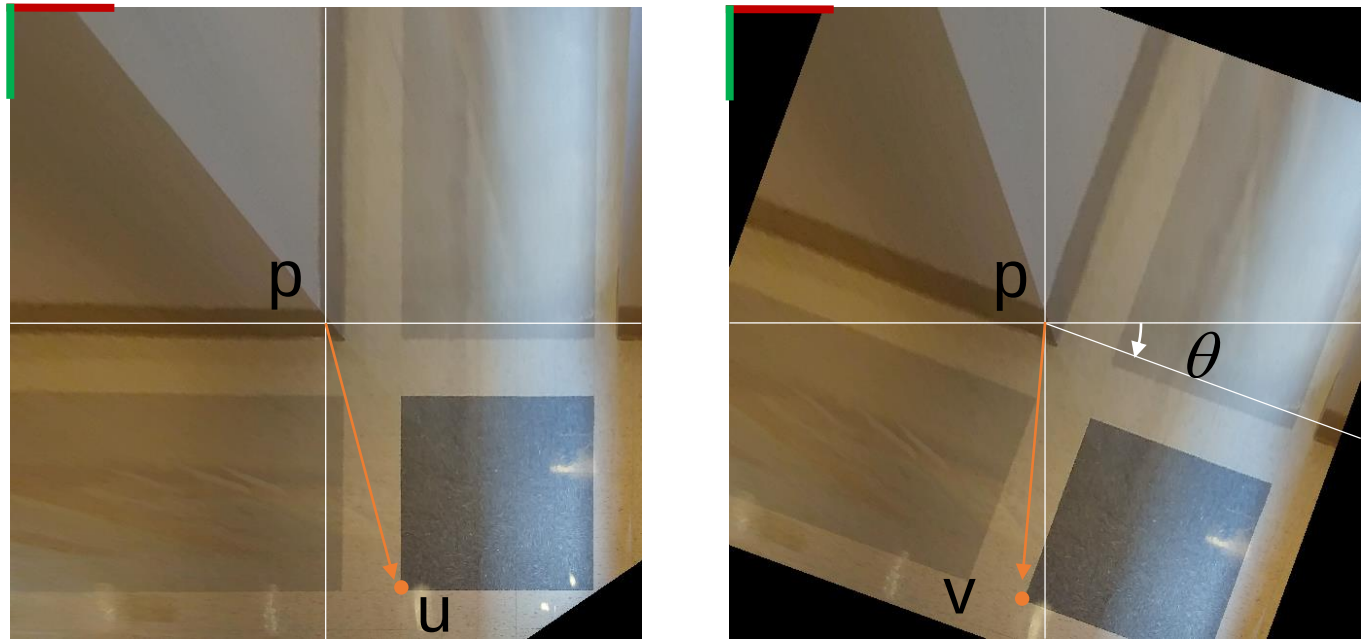
$$\bar{u} = u - p \qquad \bar{v} = v - p$$

$$\longrightarrow \bar{v} = R\bar{u}$$

$$\longrightarrow v - p = R(u - p)$$

$$\longrightarrow v = Ru - Rp + p$$

Euclidean Transform SE(2)



Rotate about the image center

$$\begin{bmatrix} v \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \longrightarrow v = Ru + t$$

$$\bar{u} = u - p \qquad \bar{v} = v - p$$

$$\longrightarrow \bar{v} = R\bar{u}$$

$$\longrightarrow v - p = R(u - p)$$

$$\longrightarrow v = Ru - Rp + p$$

$$\longrightarrow t = -Rp + p$$

Euclidean Transform SE(2)

```
im = imread('rect.png');

theta = 20/180*pi;

R = [cos(theta) -sin(theta);
     sin(theta) cos(theta)];
p = [size(im,2)/2; size(im,1)/2];
```

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Rotation around the center of image

Euclidean Transform SE(2)

RectificationViaEuclidean.m

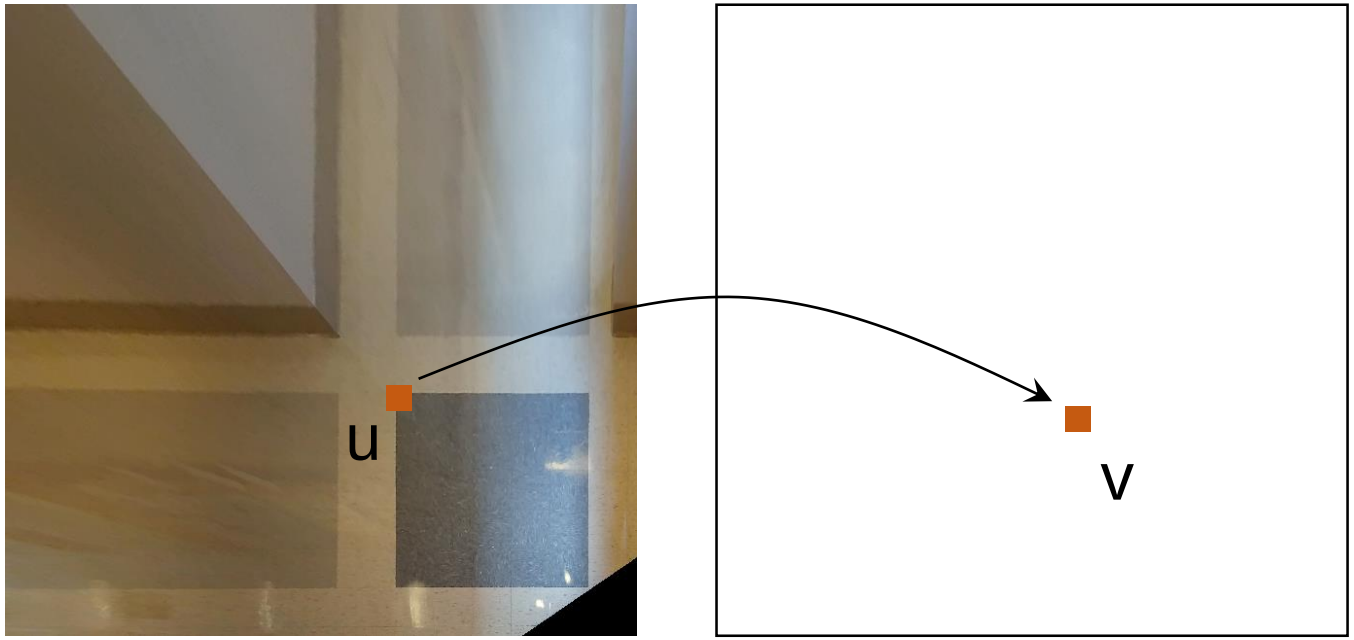
```
im = imread('rect.png');

theta = 20/180*pi;

R = [cos(theta) -sin(theta);
     sin(theta) cos(theta)];
p = [size(im,2)/2; size(im,1)/2];

T = [R -R*t+t;0 0 1];

im_warped = ImageWarpingEuclidean(im, T);
```

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} R & -Rp+p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Rotation around the center of image

Euclidean Transform SE(2)

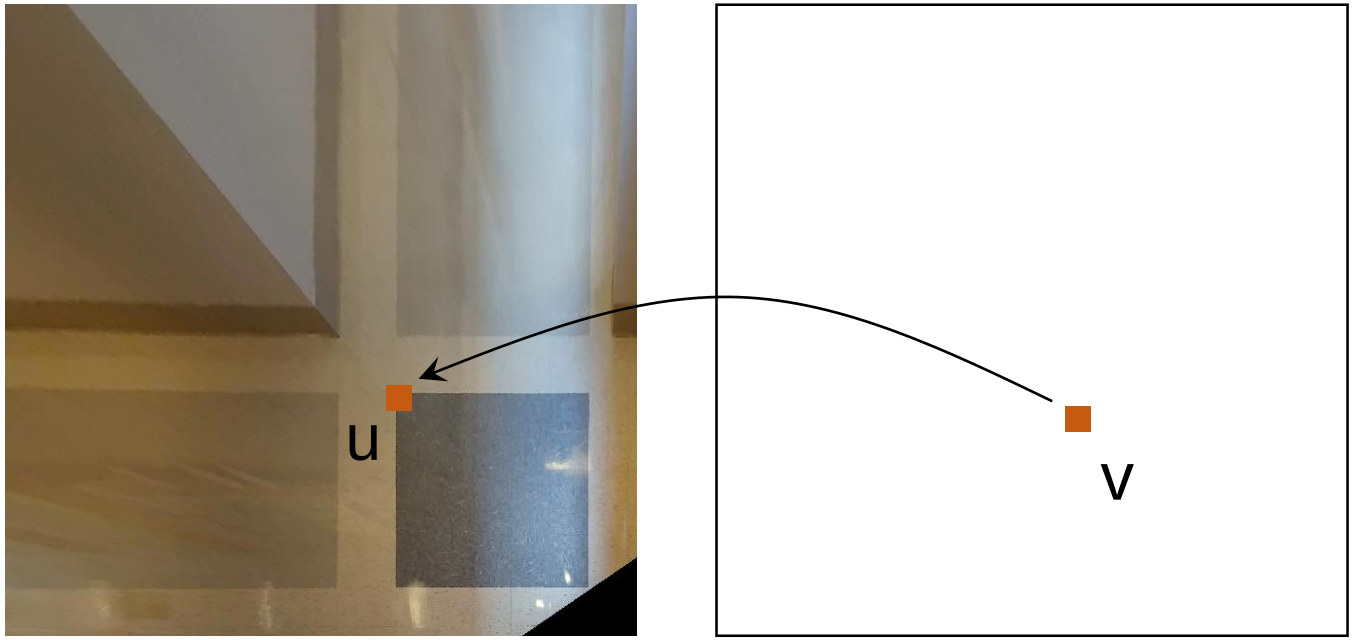


$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = H \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

ImageWarpingEuclidean.m

```
function im_warped = ImageWarpingEuclidean(im, H)

im = double(im);
```

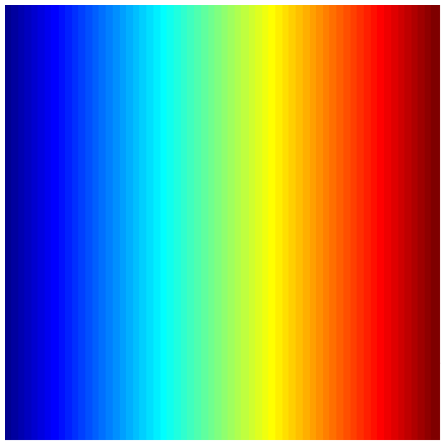
Euclidean Transform SE(2)



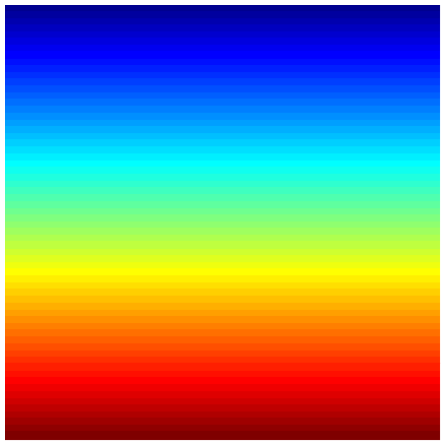
$$H^{1}\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

ImageWarpingEuclidean.m

```
function im_warped = ImageWarpingEuclidean(im, H)

im = double(im);

H = inv(H);

[v_x, v_y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(v_x, 1); w = size(v_x,2);
```

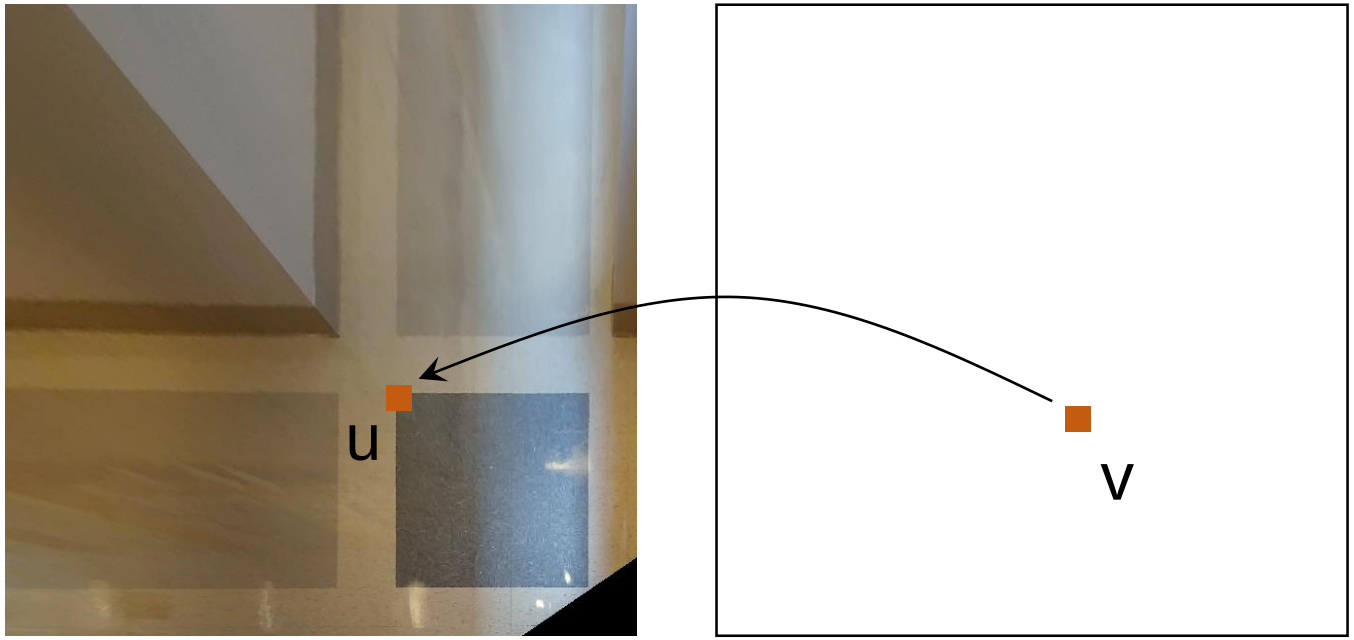


v_x



v_y

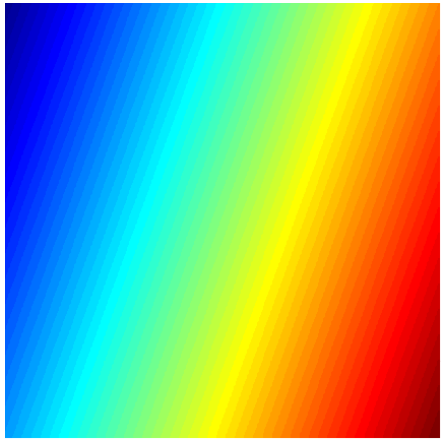
Euclidean Transform SE(2)



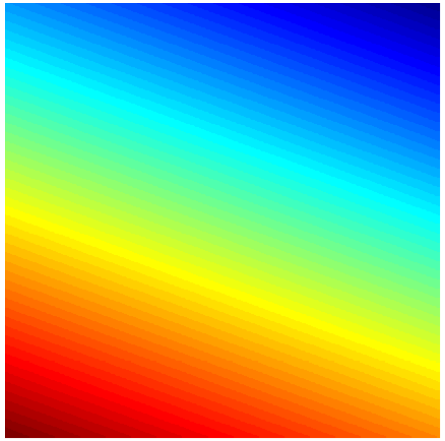
$$H^{1}\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

ImageWarpingEuclidean.m

```
function im_warped = ImageWarpingEuclidean(im, H)

im = double(im);

H = inv(H);

[v_x, v_y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(v_x, 1); w = size(v_x,2);

u_x = H(1,1)*v_x + H(1,2)*v_y + H(1,3);
u_y = H(2,1)*v_x + H(2,2)*v_y + H(2,3);
```

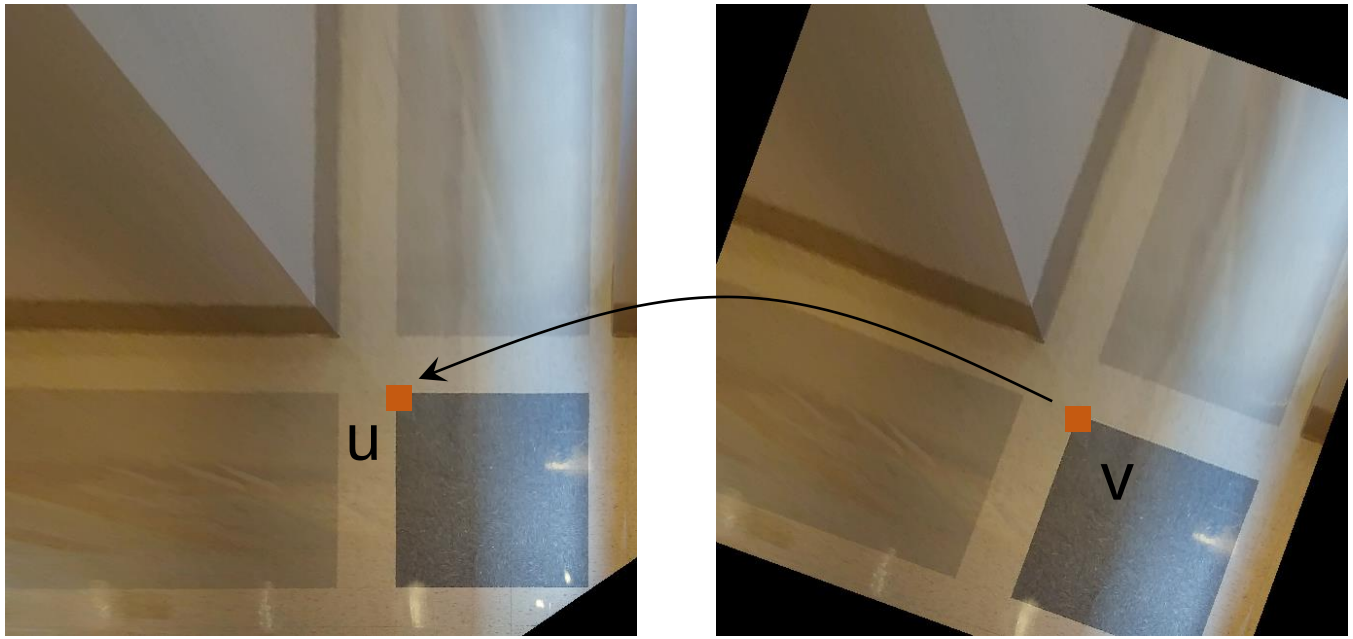


u_x



u_y

Euclidean Transform SE(2)

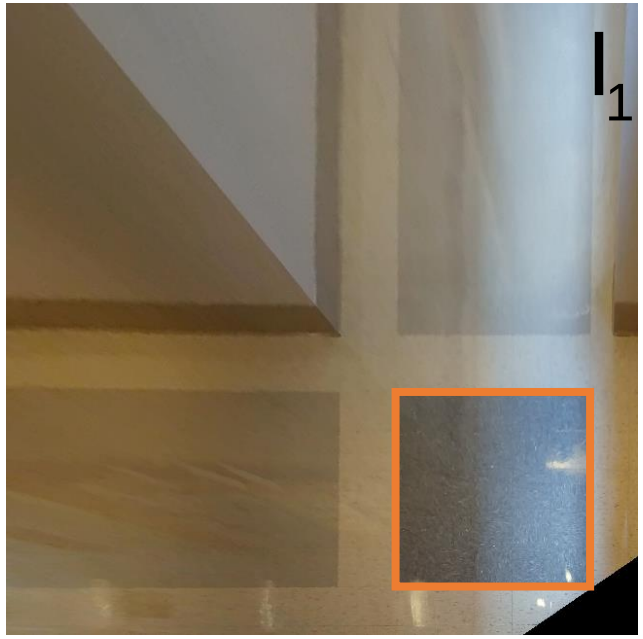


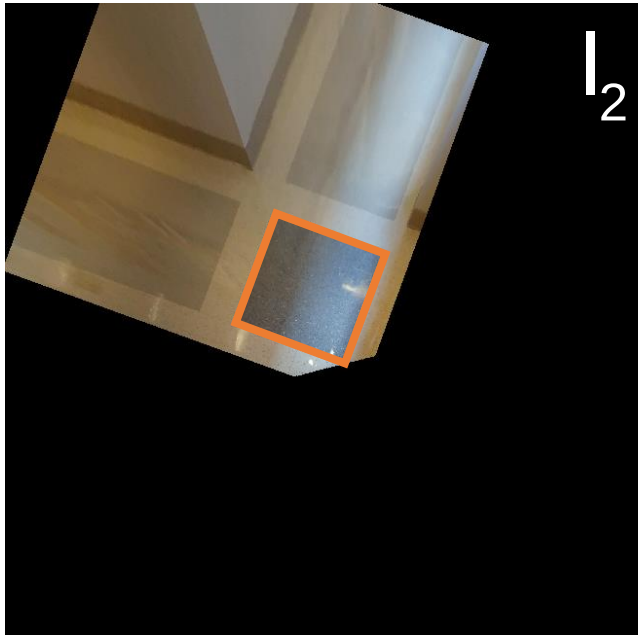
$$H^1 \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

ImageWarpingEuclidean.m

```
function im_warped = ImageWarpingEuclidean(im, H)

im = double(im);

H = inv(H);

[v_x, v_y] = meshgrid(1:(size(im,2)), 1:(size(im,1)));
h = size(v_x, 1); w = size(v_x,2);

u_x = H(1,1)*v_x + H(1,2)*v_y + H(1,3);
u_y = H(2,1)*v_x + H(2,2)*v_y + H(2,3);

im_warped(:,:,1) = reshape(interp2(im(:,:,1), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,2) = reshape(interp2(im(:,:,2), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,3) = reshape(interp2(im(:,:,3), u_x(:), u_y(:)), [h, w]);

im_warped = uint8(im_warped);
```

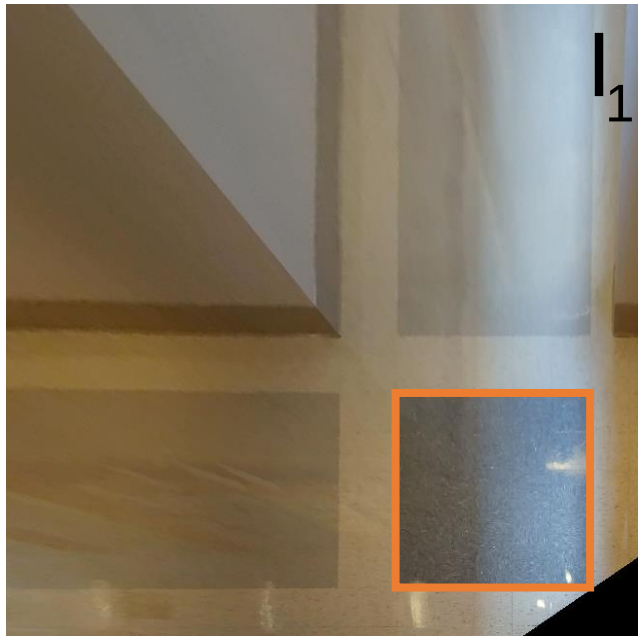
Similarity Transform

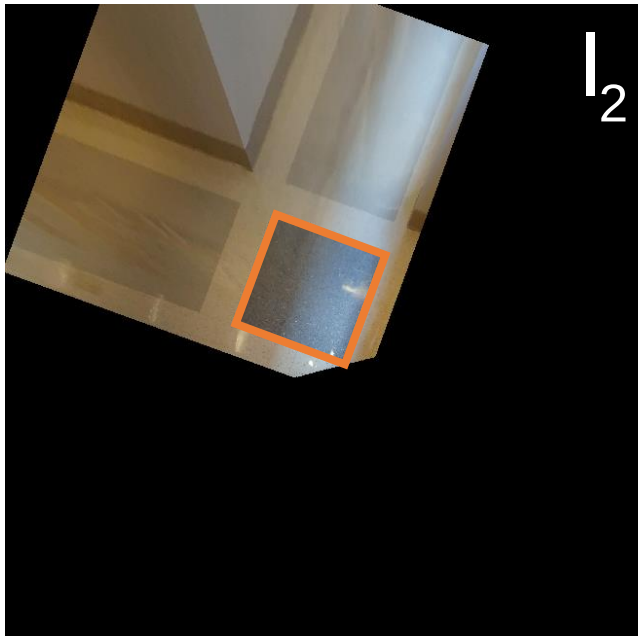




$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & & \\ & \alpha & \\ & & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Similarity Transform





Invariant properties

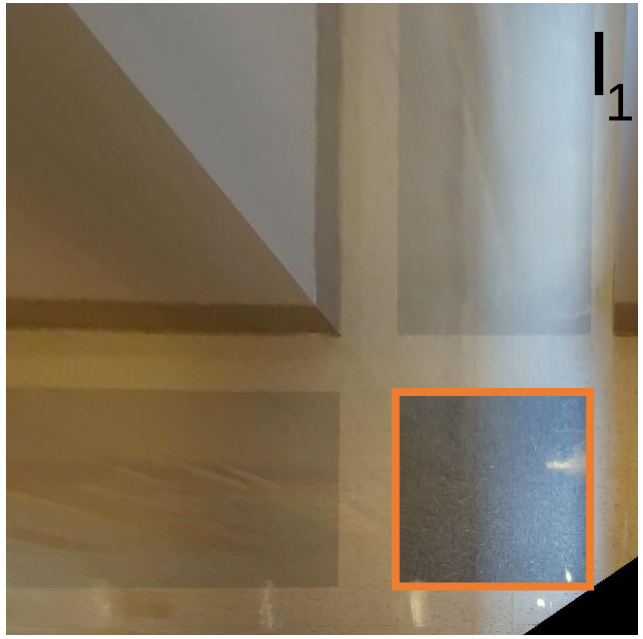
- Length ratio
- Angle

Degree of freedom

4 (2 translation+1 rotation+1 scale)

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & & \\ & \alpha & \\ & & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ & & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

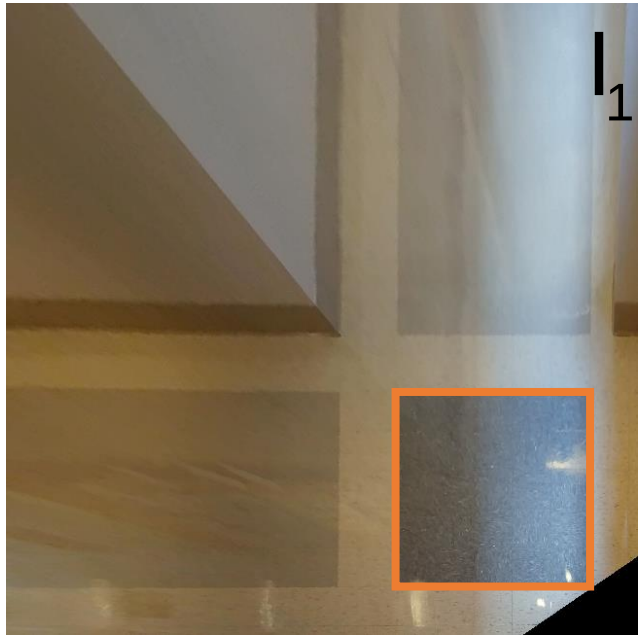
Affine Transform



$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Euclidean transform

Affine Transform

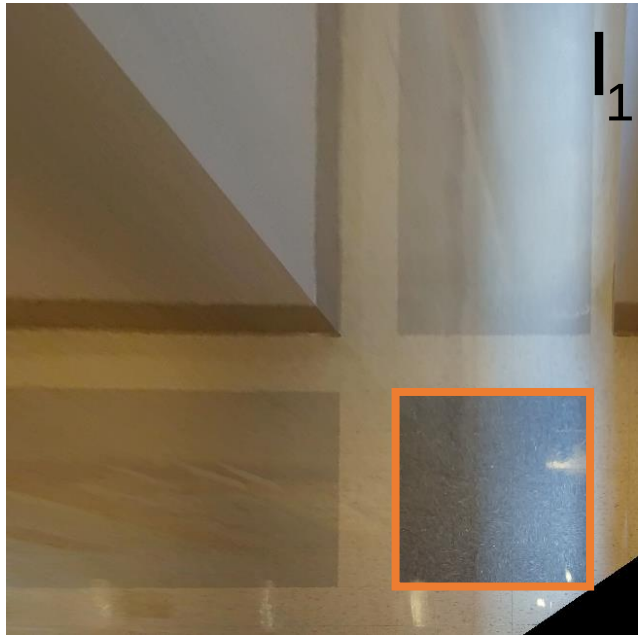


$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Euclidean transform

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Affine Transform



Invariant properties

- Parallelism
- Ratio of area
- Ratio of length

Degree of freedom

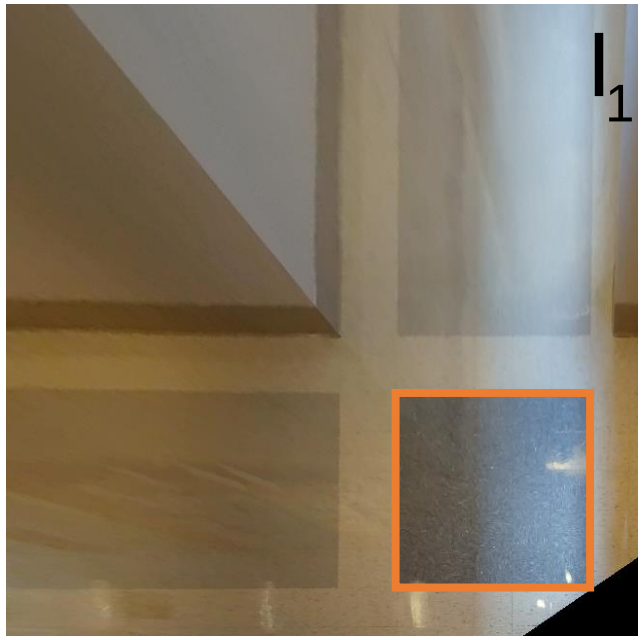
6

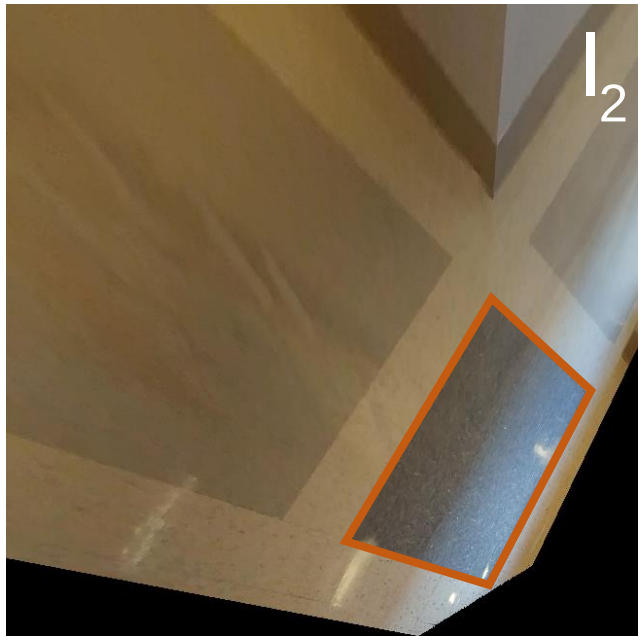
$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Euclidean transform

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

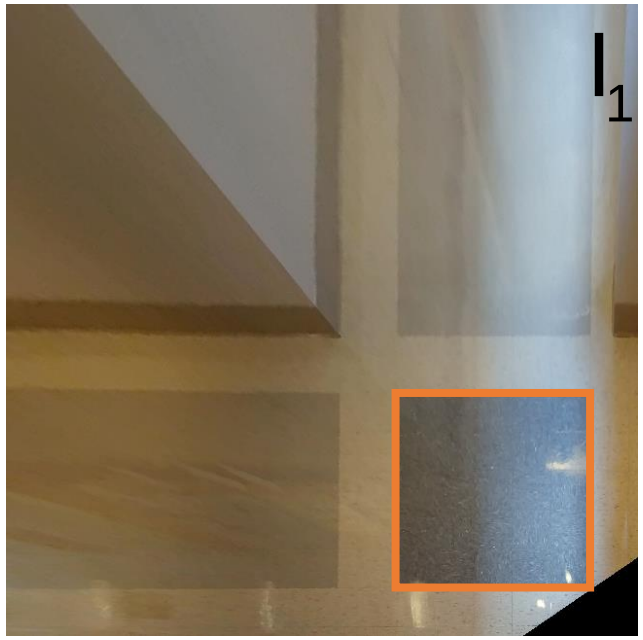
Perspective Transform (Homography)

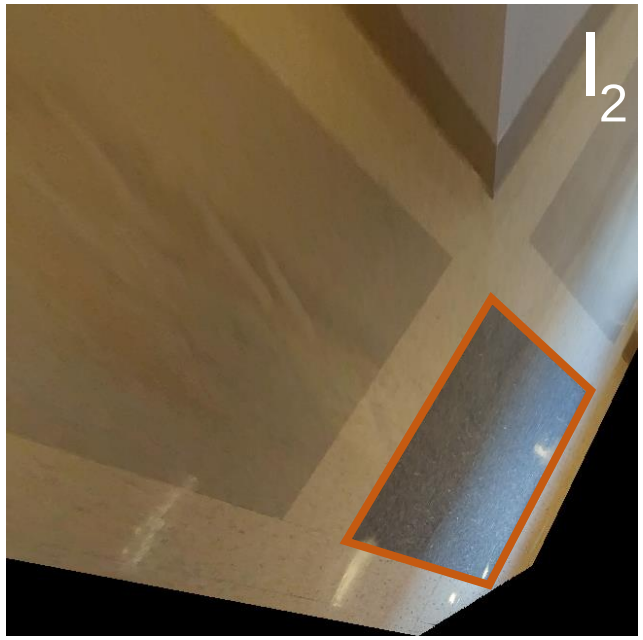




$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Perspective Transform (Homography)



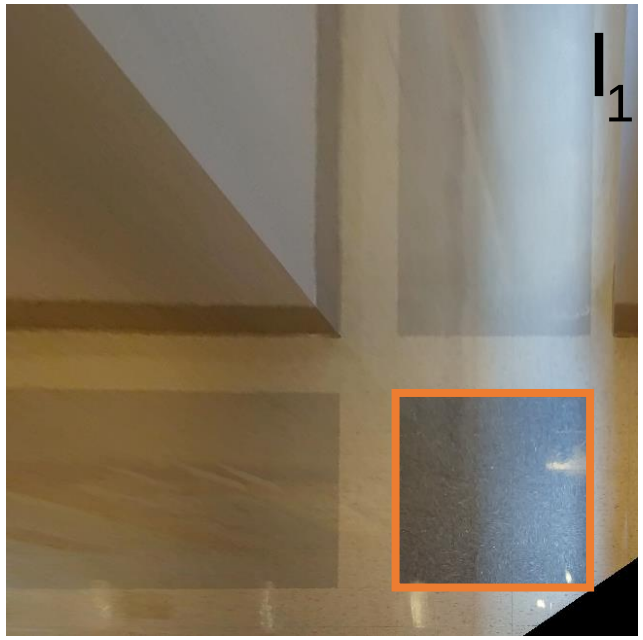


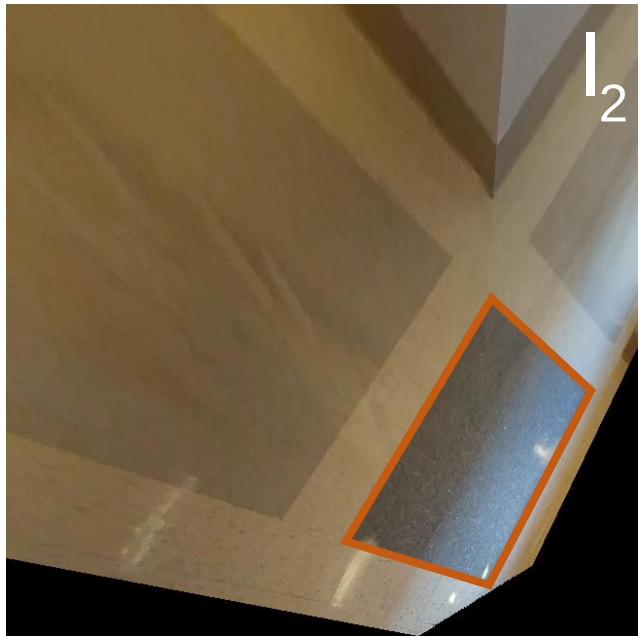
$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \mathrm{H} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

: General form of plane to plane linear mapping

Video 5.4
Jianbo Shi

Perspective Transform (Homography)

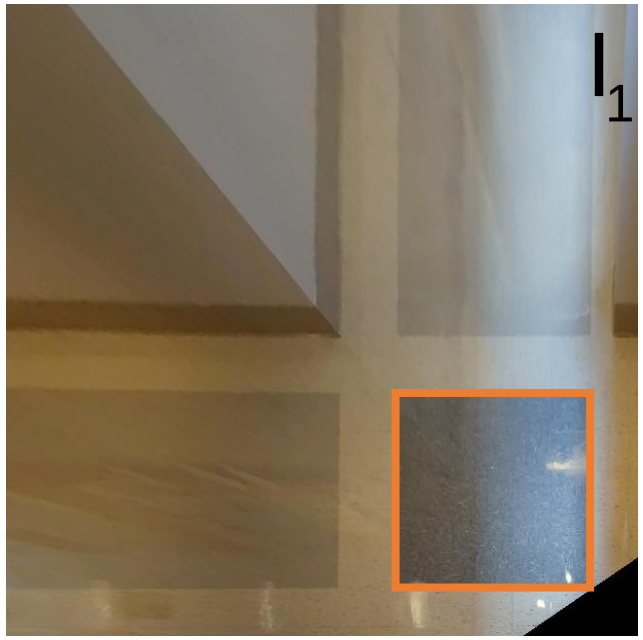


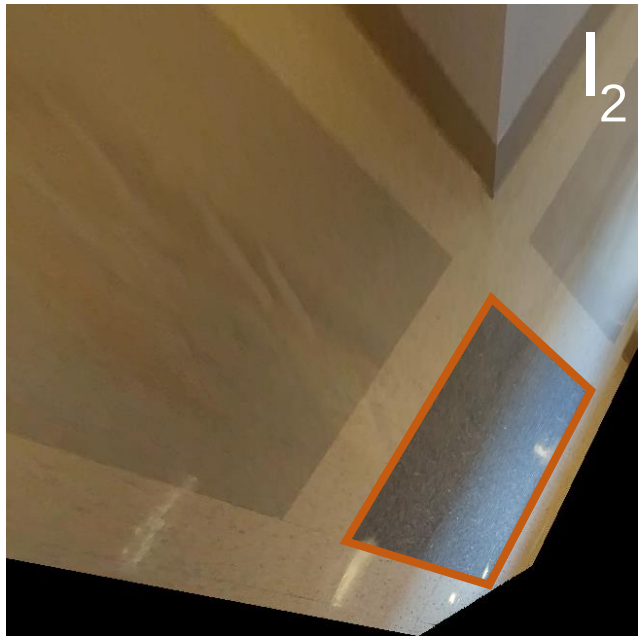


$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \mathrm{H} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

: General form of plane to plane linear mapping

Perspective Transform (Homography)

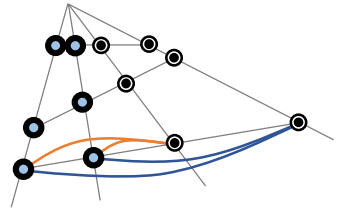




$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix} = \mathrm{H} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

Invariant properties

- Cross ratio



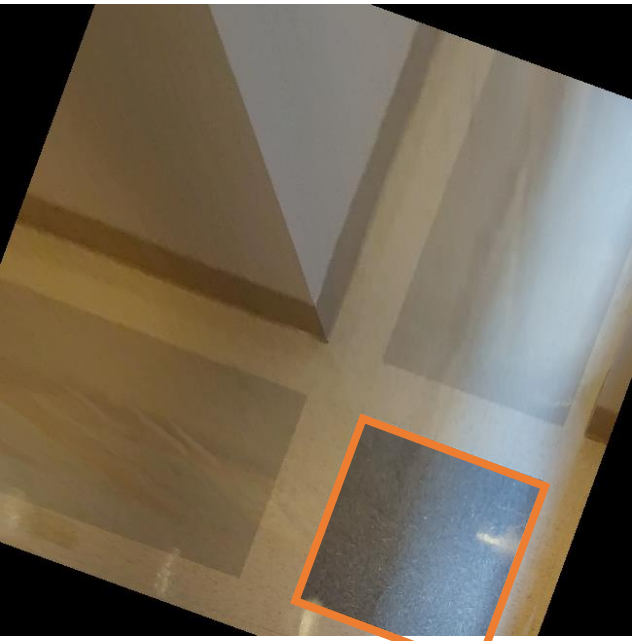
- Concurrency

- Colinearity

Degree of freedom

8 (9 variables – 1 scale)

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)

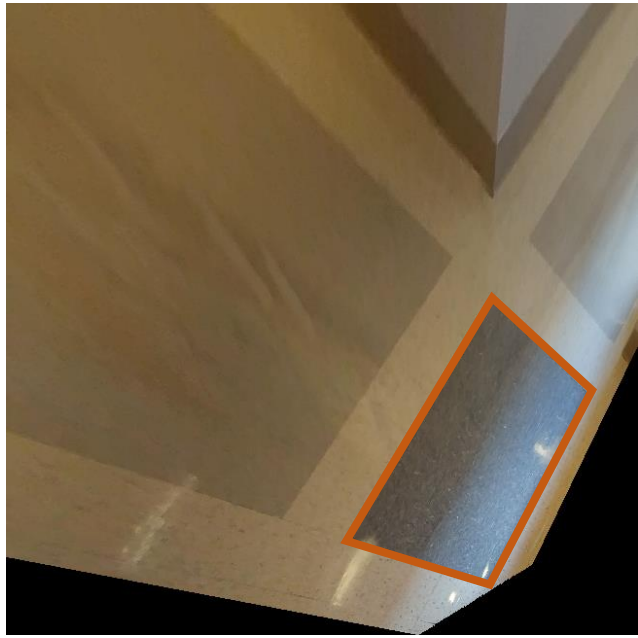
- Length ratio
- Angle

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

Affine (6 dof)

- Parallelism
- Ratio of area
- Ratio of length

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

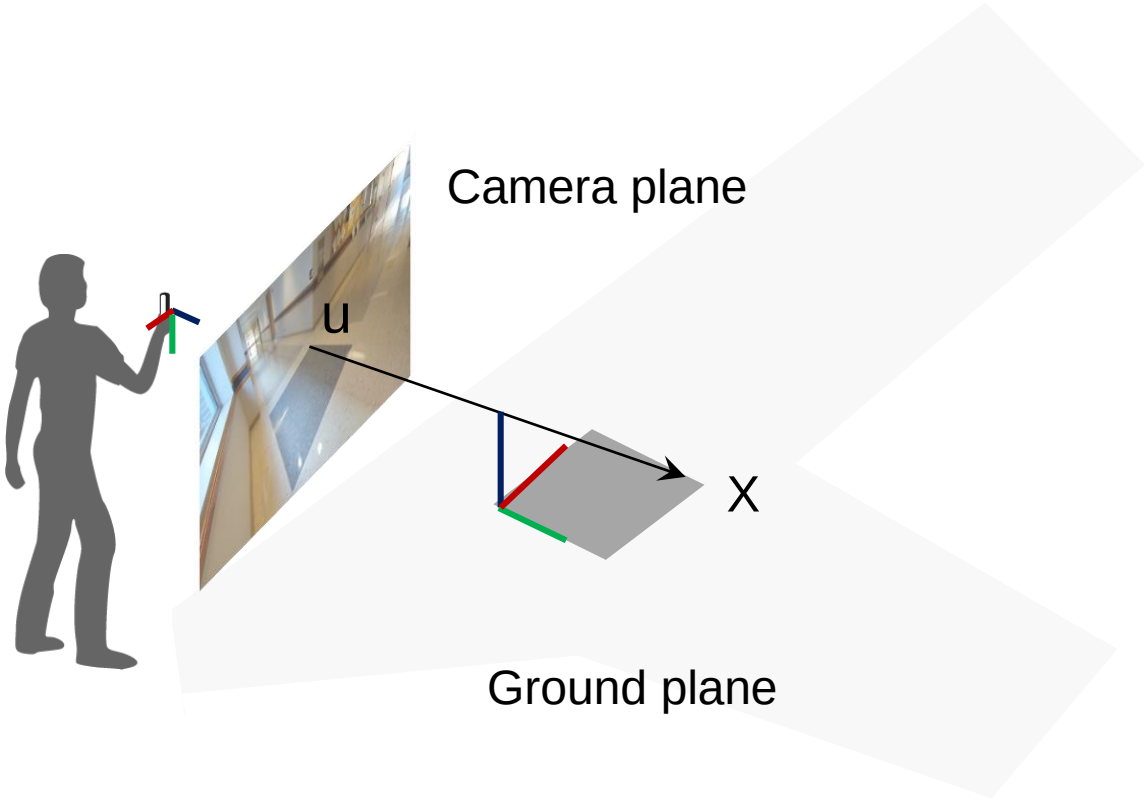


Projective (8 dof)

- Cross ratio
- Concurrency
- Colinearity

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

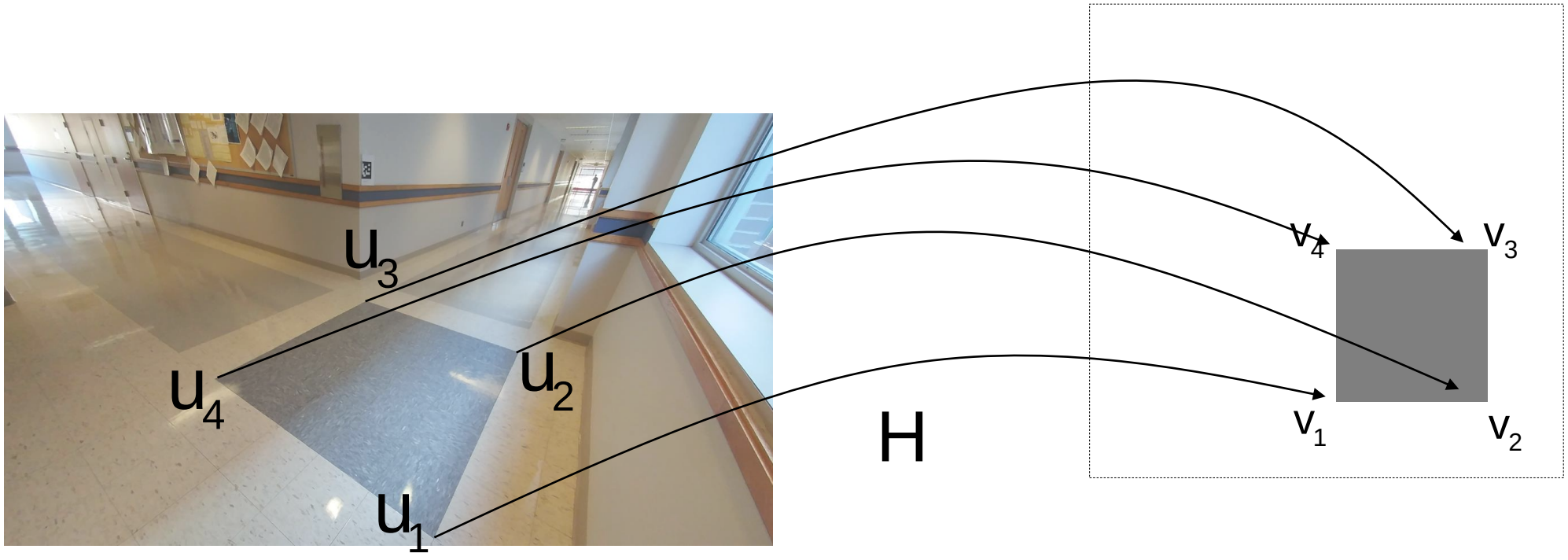
Perspective Transform (Homography)



$$\lambda \mathbf{u} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

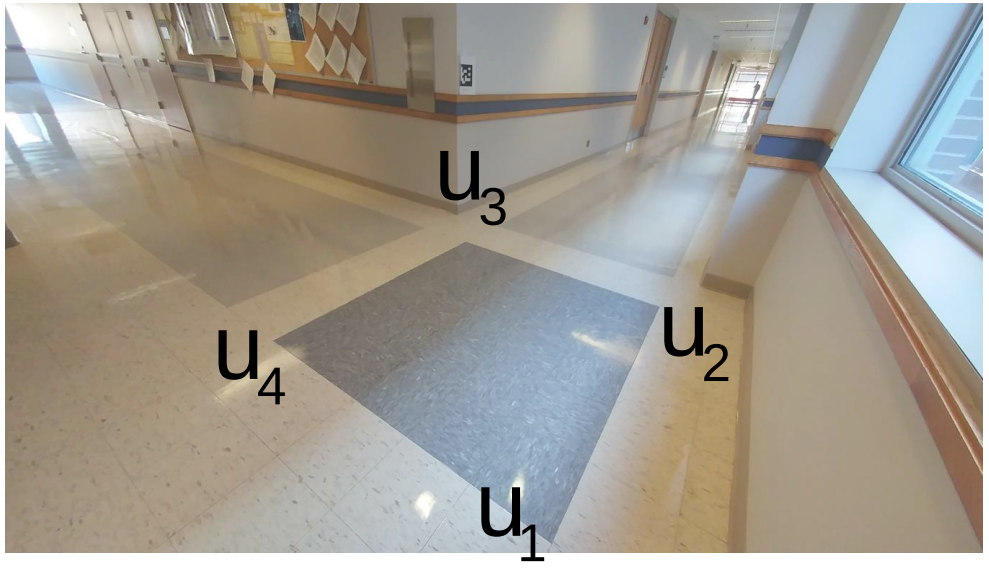
Camera plane Ground plane

Fun with Homography



The image can be rectified as if it is seen from top

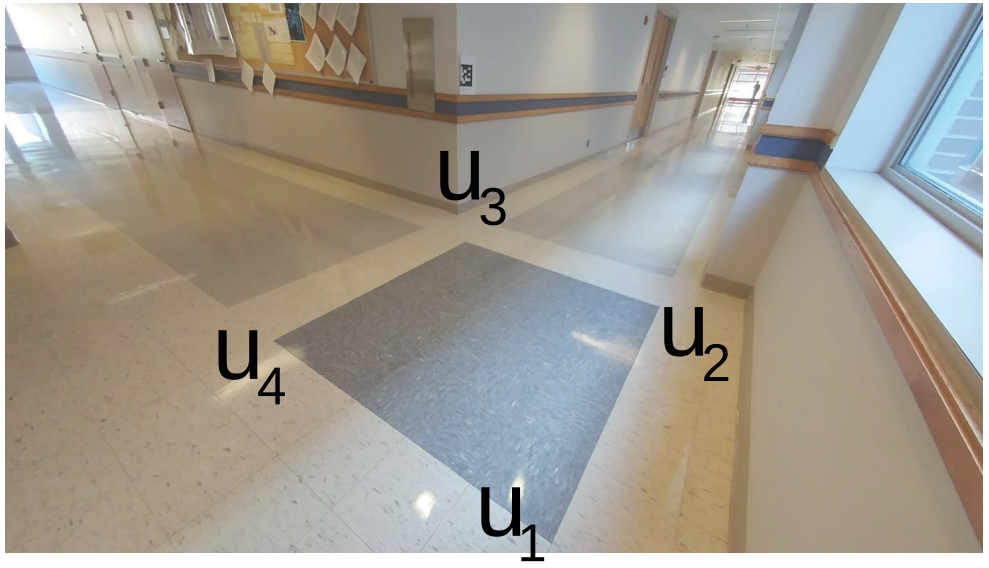
Fun with Homography



RectificationViaHomography.m

```
u = [u1'; u2'; u3'; u4'];
v = [v1'; v2'; v3'; v4'];

% Need at least non-colinear four points
H = ComputeHomography(v, u);

im_warped = ImageWarping(im, inv(H));
```

Fun with Homography



RectificationViaHomography.m

```
u = [u1'; u2'; u3'; u4'];
v = [v1'; v2'; v3'; v4'];

% Need at least non-colinear four points
H = ComputeHomography(v, u);

im_warped = ImageWarping(im, inv(H));
```

ImageWarping.m

```
u_x = H(1,1)*v_x + H(1,2)*v_y + H(1,3);
u_y = H(2,1)*v_x + H(2,2)*v_y + H(2,3);
u_z = H(3,1)*v_x + H(3,2)*v_y + H(3,3);

u_x = u_x./u_z;
u_y = u_y./u_z;
```

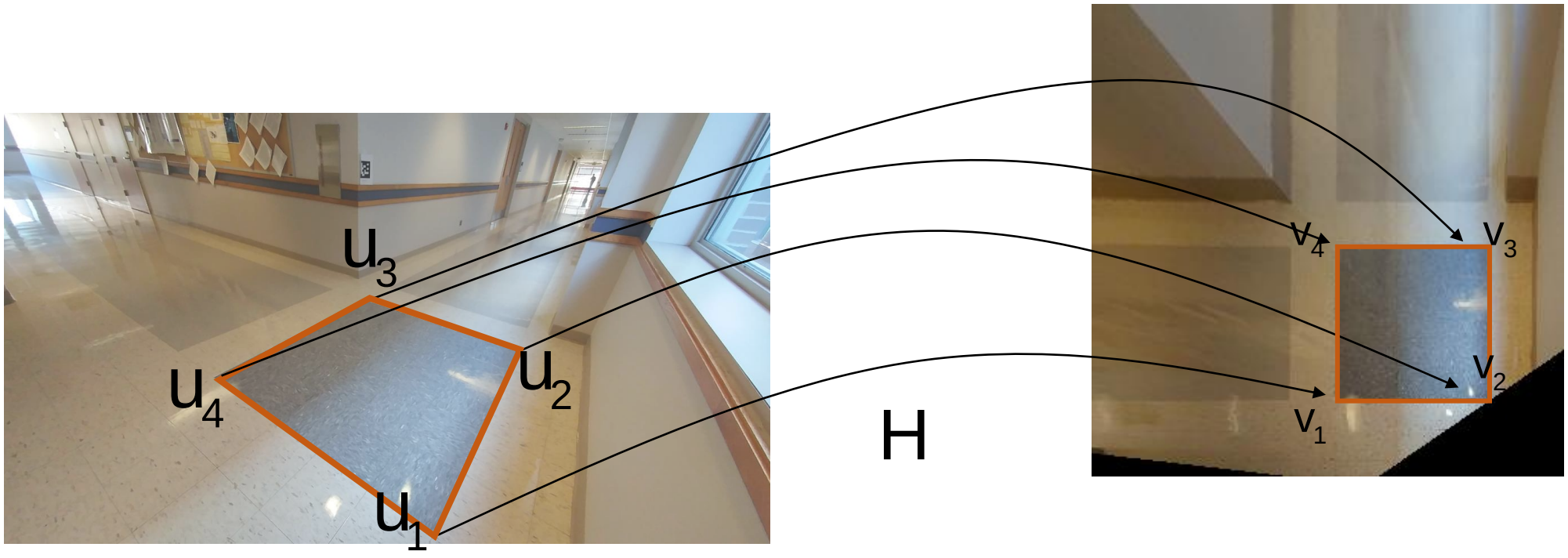
$$\leftarrow \lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

```
im_warped(:,:,1) = reshape(interp2(im(:,:,1), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,2) = reshape(interp2(im(:,:,2), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,3) = reshape(interp2(im(:,:,3), u_x(:), u_y(:)), [h, w]);

im_warped = uint8(im_warped);
```

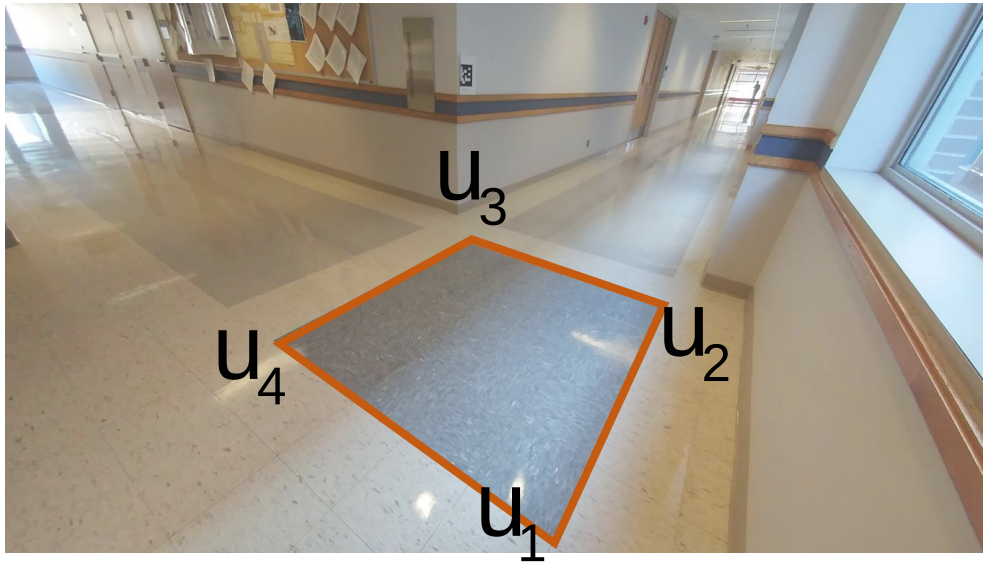
Cf) ImageWarpingEuclidean.m

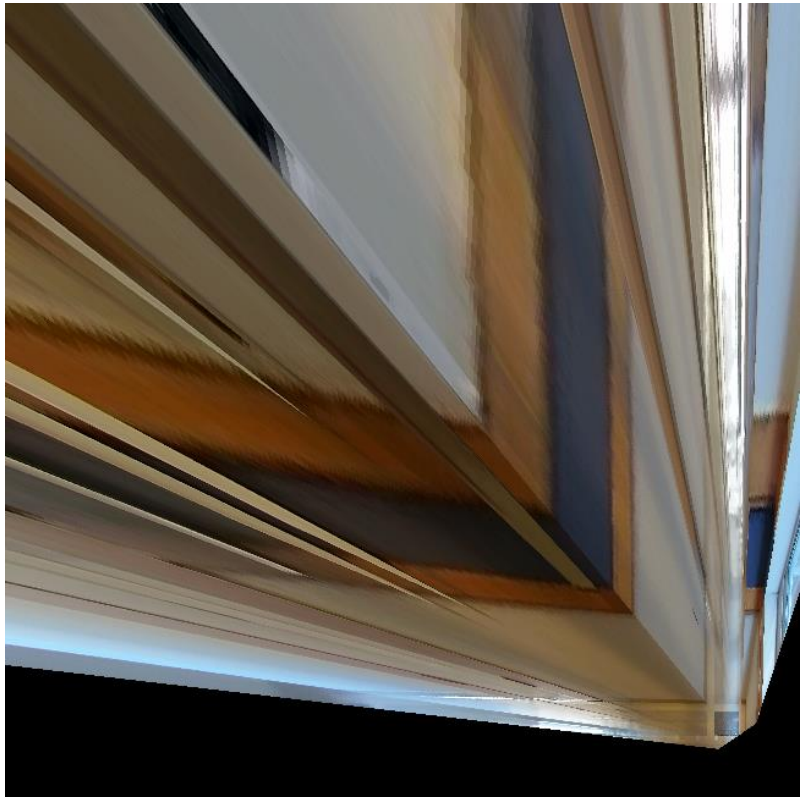
```
u_x = H(1,1)*v_x + H(1,2)*v_y + H(1,3);
u_y = H(2,1)*v_x + H(2,2)*v_y + H(2,3);
```

Fun with Homography



Fun with Homography





Fun with Homography

Fun with Homography

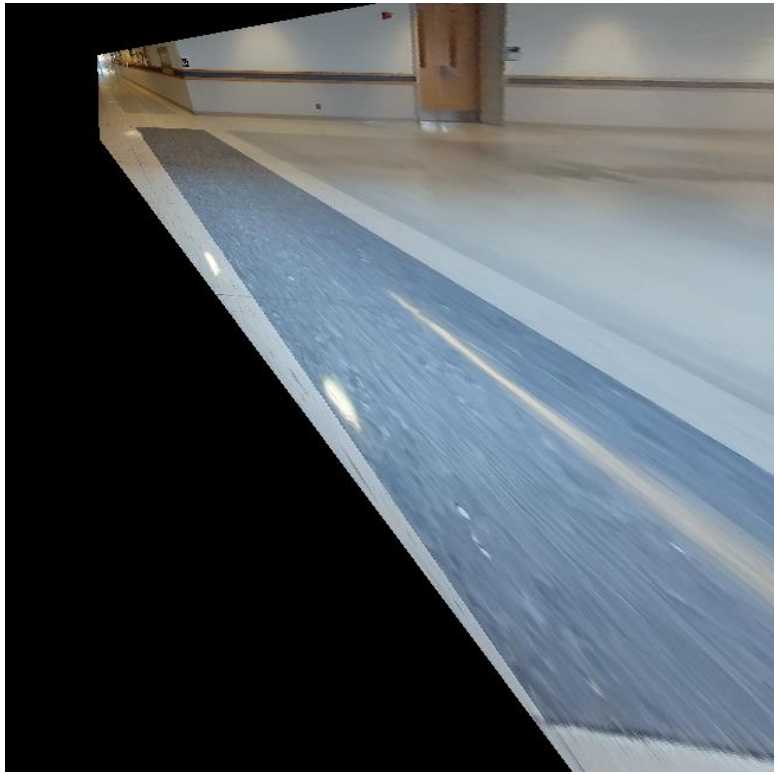


Image Inpainting

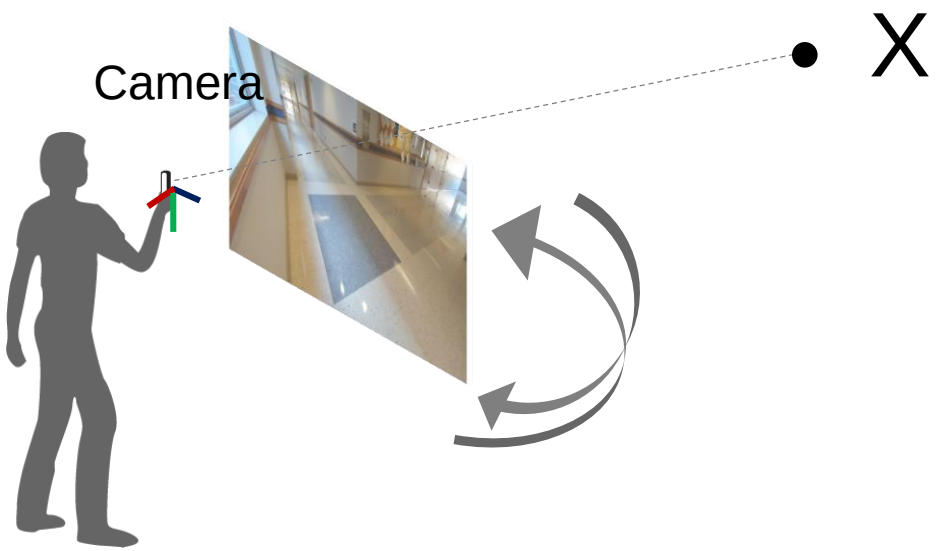
Image Inpainting

Virtual Advertisement

360 Panorama

https://www.youtube.com/watch?v=H6SsB3JYqC

Image Transform by Pure 3D Rotation



$$\lambda u = Hv$$

Left

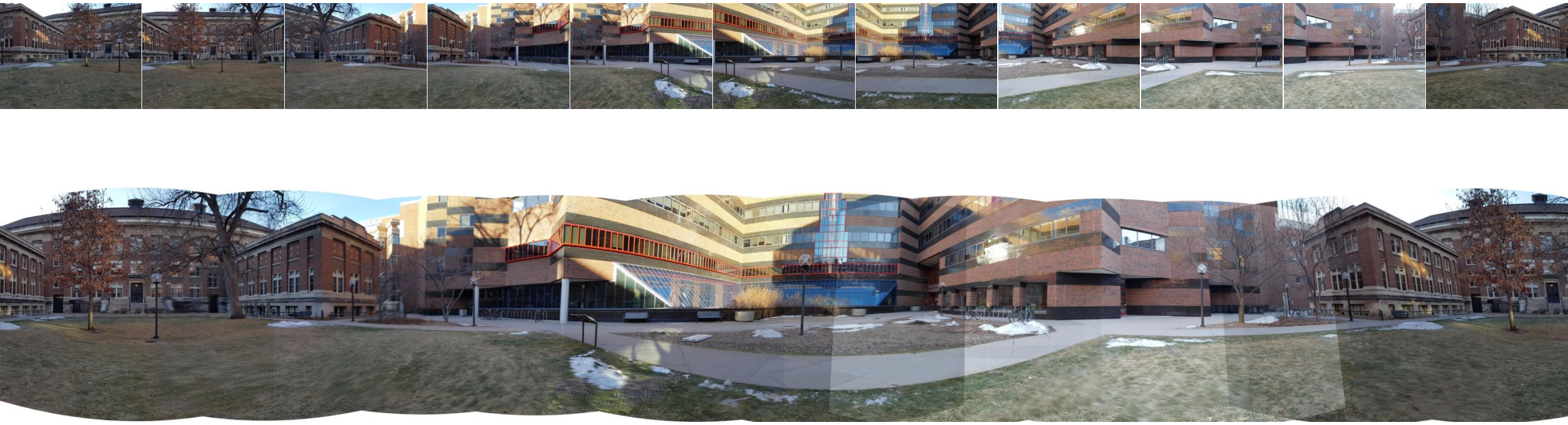
Right

Euclidean Transform (Translation)


Homography

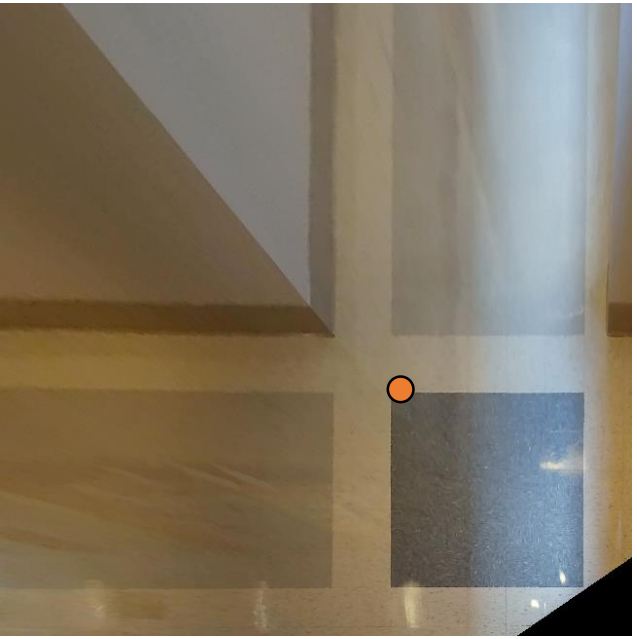
Image Panorama (Cylindrical Projection)

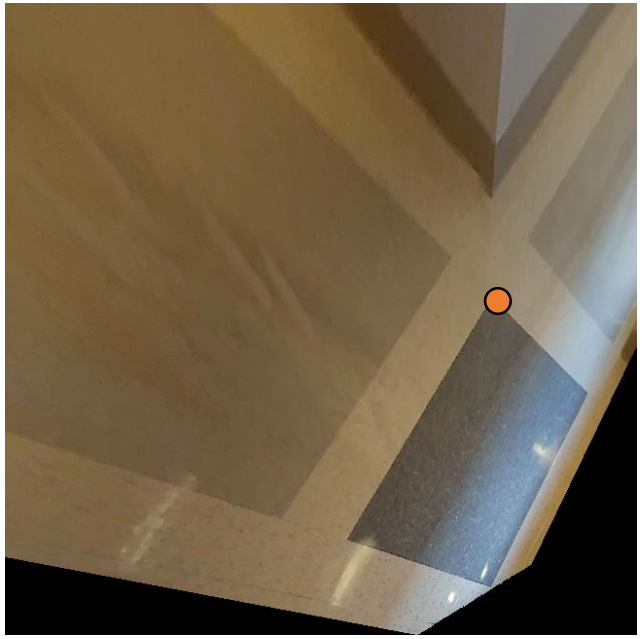
Image Panorama (Cylindrical Projection)



Video 5.5
Jianbo Shi

Homography Computation



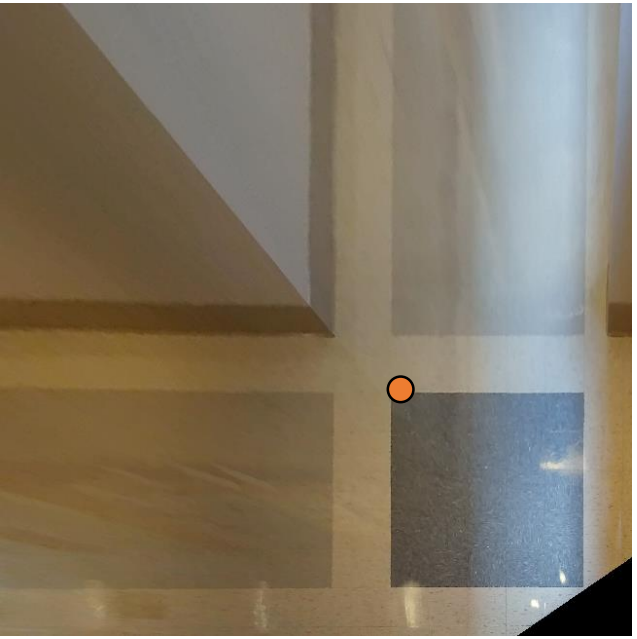


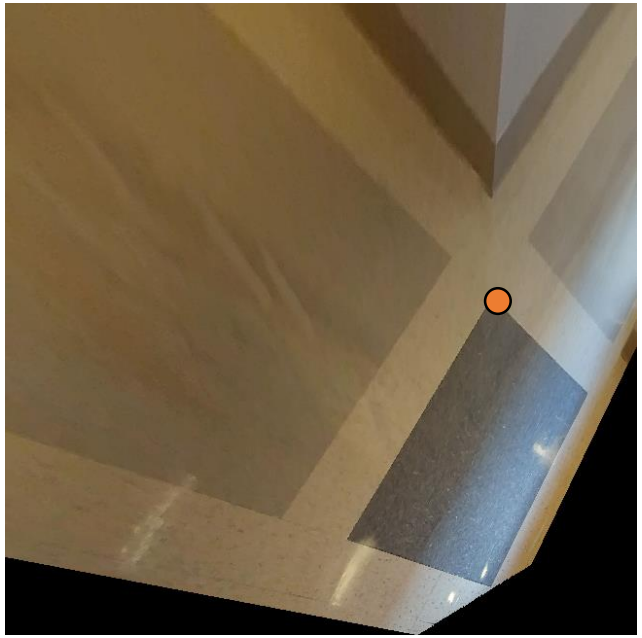
$$v_x = \frac{h_{11}u_x + h_{12}u_y + h_{13}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

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

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

Homography Computation





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

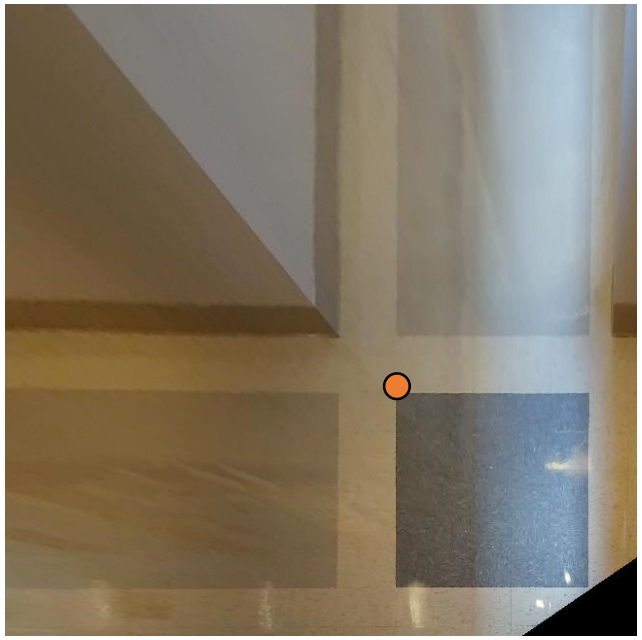
$$v_y = \frac{h_{21}u_x + h_{22}u_y + h_{23}}{h_{31}u_x + h_{32}u_y + h_{33}}$$

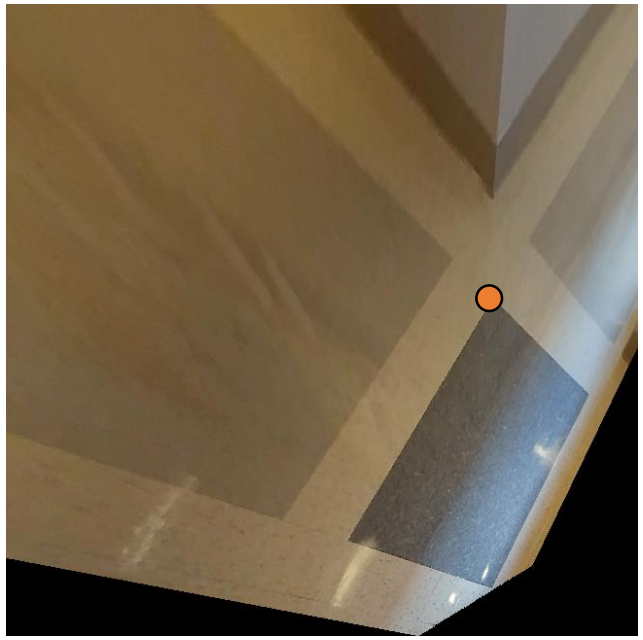
$$\rightarrow \begin{aligned} h_{11}u_x + h_{12}u_y + h_{13} - h_{31}u_x v_x - h_{32}u_y v_x - h_{33}v_x = 0 \\ h_{21}u_x + h_{22}u_y + h_{23} - h_{31}u_x v_y - h_{32}u_y v_y - h_{33}v_y = 0 \end{aligned}$$

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

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

Homography Computation





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

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

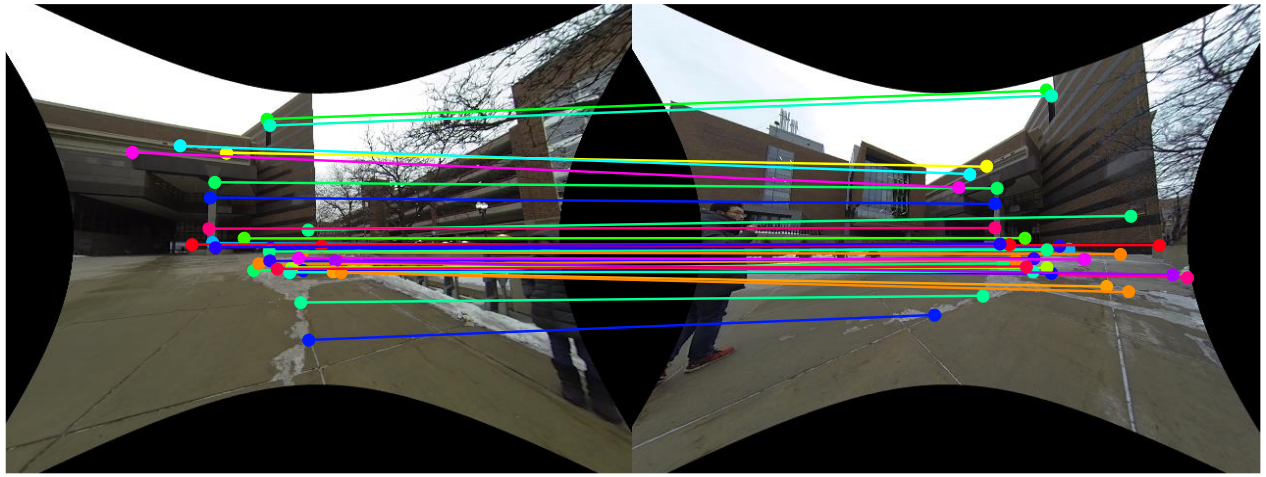
$$\rightarrow \quad h_{11}u_x + h_{12}u_y + h_{13} - h_{31}u_x v_x - h_{32}u_y v_x - h_{33}v_x = 0$$
$$h_{21}u_x + h_{22}u_y + h_{23} - h_{31}u_x v_y - h_{32}u_y v_y - h_{33}v_y = 0$$

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

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

2x9

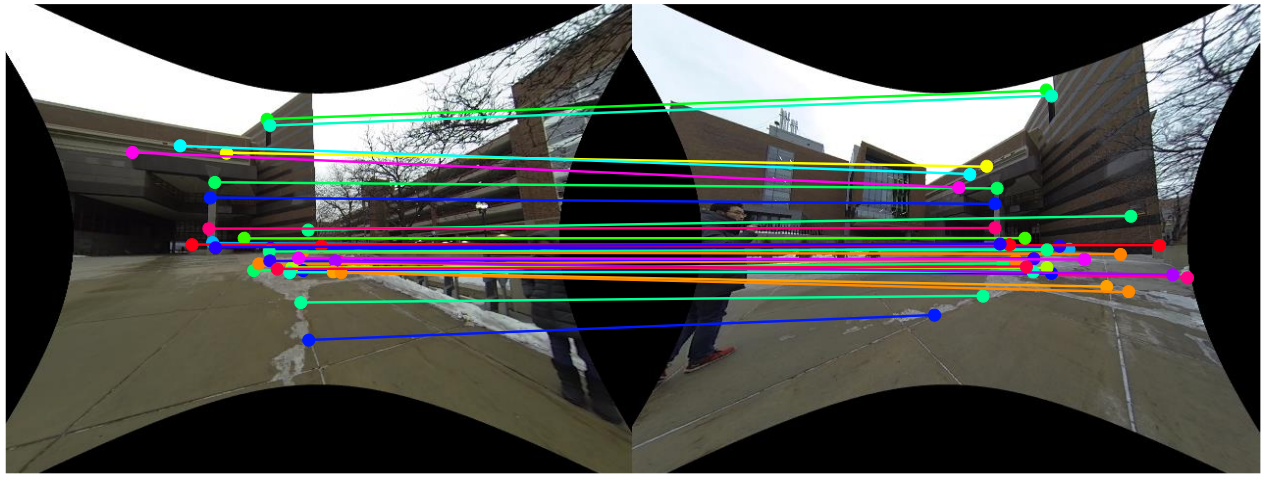
Linear System for Homography Matrix



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

2x9

How Many Correspondences?



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

2x9

$$\mathbf{x} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}$$

What is minimum m?

I_1

$$\left\{\begin{array}{l} v_1 \leftrightarrow u_1 \\ v_2 \leftrightarrow u_2 \\ v_3 \leftrightarrow u_3 \\ v_4 \leftrightarrow u_4 \end{array}\right\} \rightarrow H$$

I_2

$$\begin{bmatrix} u^1_x & u^1_y & 1 & & & & -u^1_x v^1_x & -u^1_y v^1_x & -v^1_x \\ & & & u^1_x & u^1_y & 1 & -u^1_x v^1_y & -u^1_y v^1_y & -v^1_y \\ u^2_x & u^2_y & 1 & & & & -u^2_x v^2_x & -u^2_y v^2_x & -v^2_x \\ & & & u^2_x & u^2_y & 1 & -u^2_x v^2_y & -u^2_y v^2_y & -v^2_y \\ u^3_x & u^3_y & 1 & & & & -u^3_x v^3_x & -u^3_y v^3_x & -v^3_x \\ & & & u^3_x & u^3_y & 1 & -u^3_x v^3_y & -u^3_y v^3_y & -v^3_y \\ u^4_x & u^4_y & 1 & & & & -u^4_x v^4_x & -u^4_y v^4_x & -v^4_x \\ & & & u^4_x & u^4_y & 1 & -u^4_x v^4_y & -u^4_y v^4_y & -v^4_y \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Homography computation

I_1

$$\left\{\begin{array}{l} v_1 \leftrightarrow u_1 \\ v_2 \leftrightarrow u_2 \\ v_3 \leftrightarrow u_3 \\ v_4 \leftrightarrow u_4 \end{array}\right\} \rightarrow H$$

Homography computation

I_2

$$\begin{bmatrix} u^1_x & u^1_y & 1 & & & & -u^1_x v^1_x & -u^1_y v^1_x & -v^1_x \\ & & & u^1_x & u^1_y & 1 & -u^1_x v^1_y & -u^1_y v^1_y & -v^1_y \\ u^2_x & u^2_y & 1 & & & & -u^2_x v^2_x & -u^2_y v^2_x & -v^2_x \\ & & & u^2_x & u^2_y & 1 & -u^2_x v^2_y & -u^2_y v^2_y & -v^2_y \\ u^3_x & u^3_y & 1 & & & & -u^3_x v^3_x & -u^3_y v^3_x & -v^3_x \\ & & & u^3_x & u^3_y & 1 & -u^3_x v^3_y & -u^3_y v^3_y & -v^3_y \\ u^4_x & u^4_y & 1 & & & & -u^4_x v^4_x & -u^4_y v^4_x & -v^4_x \\ & & & u^4_x & u^4_y & 1 & -u^4_x v^4_y & -u^4_y v^4_y & -v^4_y \end{bmatrix} \mathbf{A} \quad \mathbf{X} \quad \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

I_1 I_2

$$\left\{\begin{array}{l} v_1 \leftrightarrow u_1 \\ v_2 \leftrightarrow u_2 \\ v_3 \leftrightarrow u_3 \\ v_4 \leftrightarrow u_4 \end{array}\right\} \rightarrow H$$

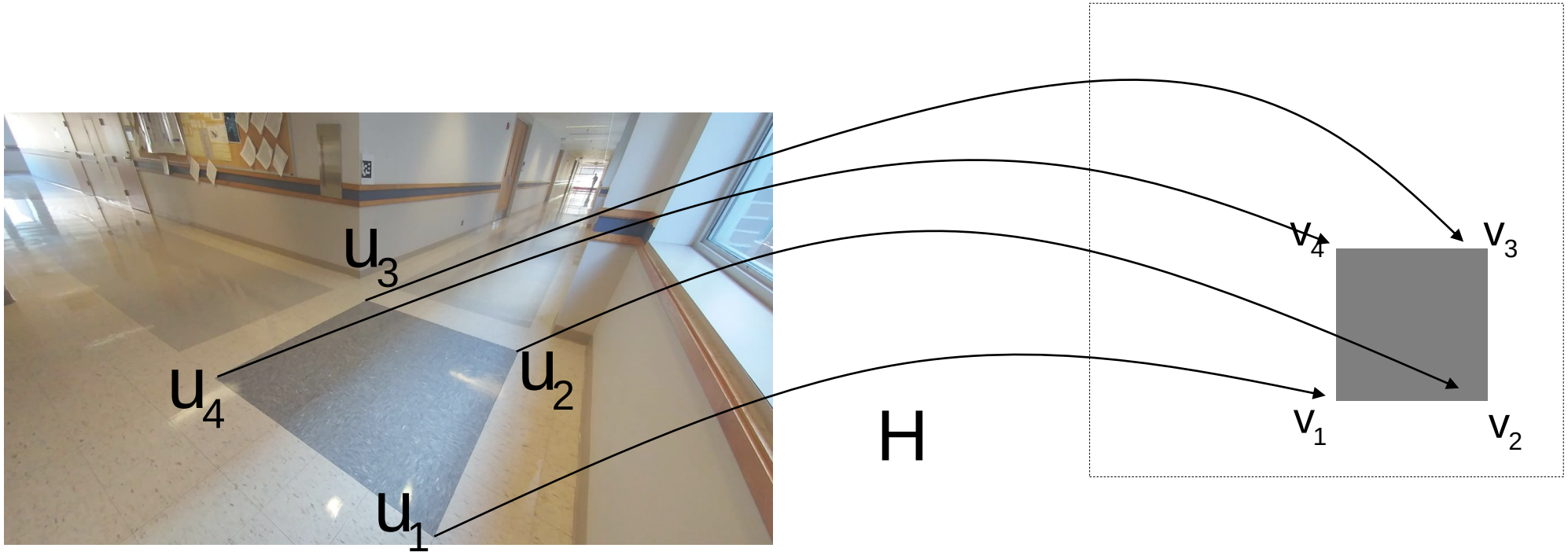
$$\begin{bmatrix} u^1_x & u^1_y & 1 & & & & -u^1_x v^1_x & -u^1_y v^1_x & -v^1_x \\ & & & u^1_x & u^1_y & 1 & -u^1_x v^1_y & -u^1_y v^1_y & -v^1_y \\ u^2_x & u^2_y & 1 & & & & -u^2_x v^2_x & -u^2_y v^2_x & -v^2_x \\ & & & u^2_x & u^2_y & 1 & -u^2_x v^2_y & -u^2_y v^2_y & -v^2_y \\ u^3_x & u^3_y & 1 & & & & -u^3_x v^3_x & -u^3_y v^3_x & -v^3_x \\ & & & u^3_x & u^3_y & 1 & -u^3_x v^3_y & -u^3_y v^3_y & -v^3_y \\ u^4_x & u^4_y & 1 & & & & -u^4_x v^4_x & -u^4_y v^4_x & -v^4_x \\ & & & u^4_x & u^4_y & 1 & -u^4_x v^4_y & -u^4_y v^4_y & -v^4_y \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

A X = 0

```
[u,d,v] = svd(A);
X =
v(:,end)/v(end,end);
H = reshape(X,3,3)';
```

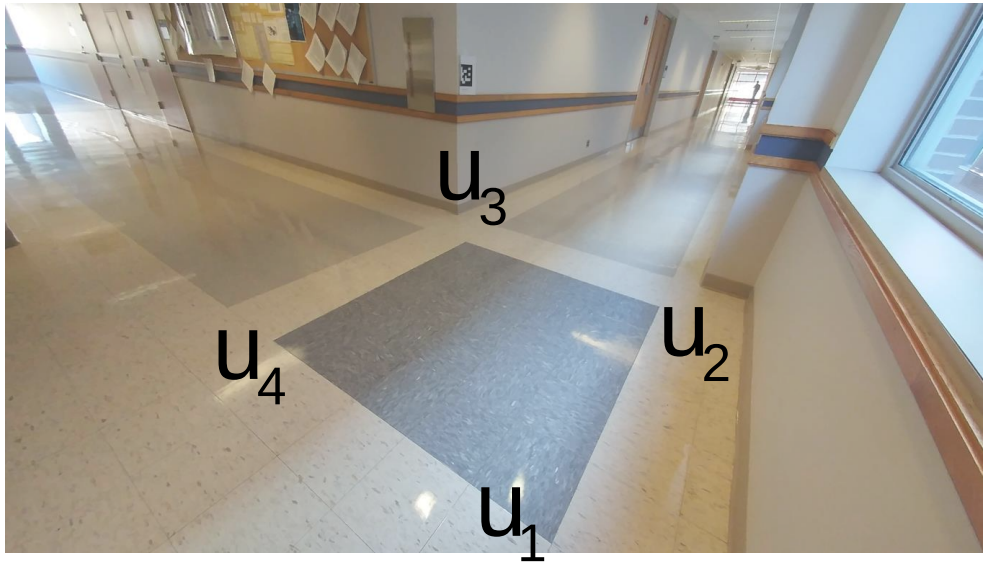
Homography computation

Fun with Homography



The image can be rectified as if it is seen from top

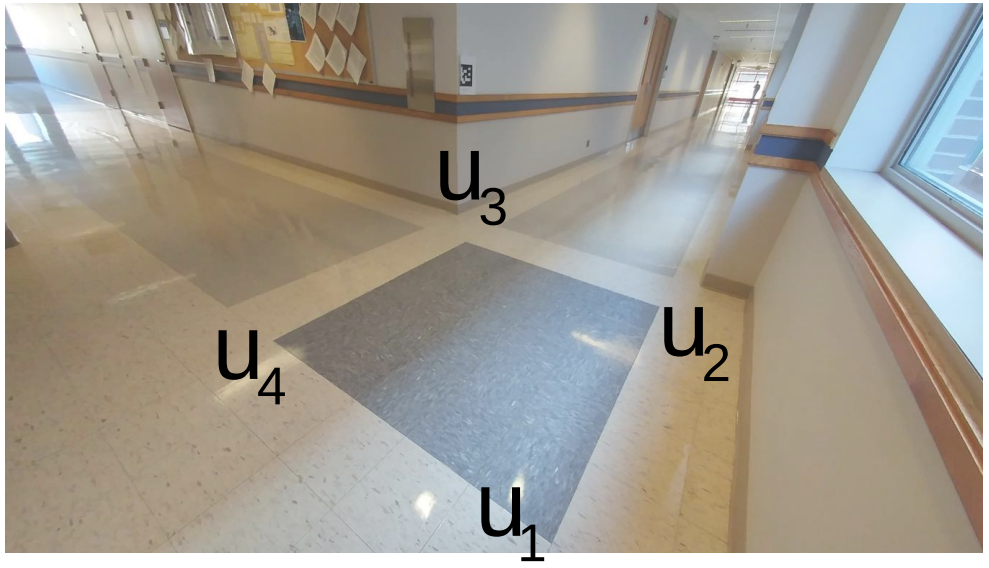
Fun with Homography



RectificationViaHomography.m

```
u = [u1'; u2'; u3'; u4'];
v = [v1'; v2'; v3'; v4'];

% Need at least non-colinear four points
H = ComputeHomography(v, u);

im_warped = ImageWarping(im, inv(H));
```

Fun with Homography



RectificationViaHomography.m

```
u = [u1'; u2'; u3'; u4'];
v = [v1'; v2'; v3'; v4'];

% Need at least non-colinear four points
H = ComputeHomography(v, u);

im_warped = ImageWarping(im, inv(H));
```

ImageWarping.m

```
u_x = H(1,1)*v_x + H(1,2)*v_y + H(1,3);
u_y = H(2,1)*v_x + H(2,2)*v_y + H(2,3);
u_z = H(3,1)*v_x + H(3,2)*v_y + H(3,3);

u_x = u_x./u_z;
u_y = u_y./u_z;
```

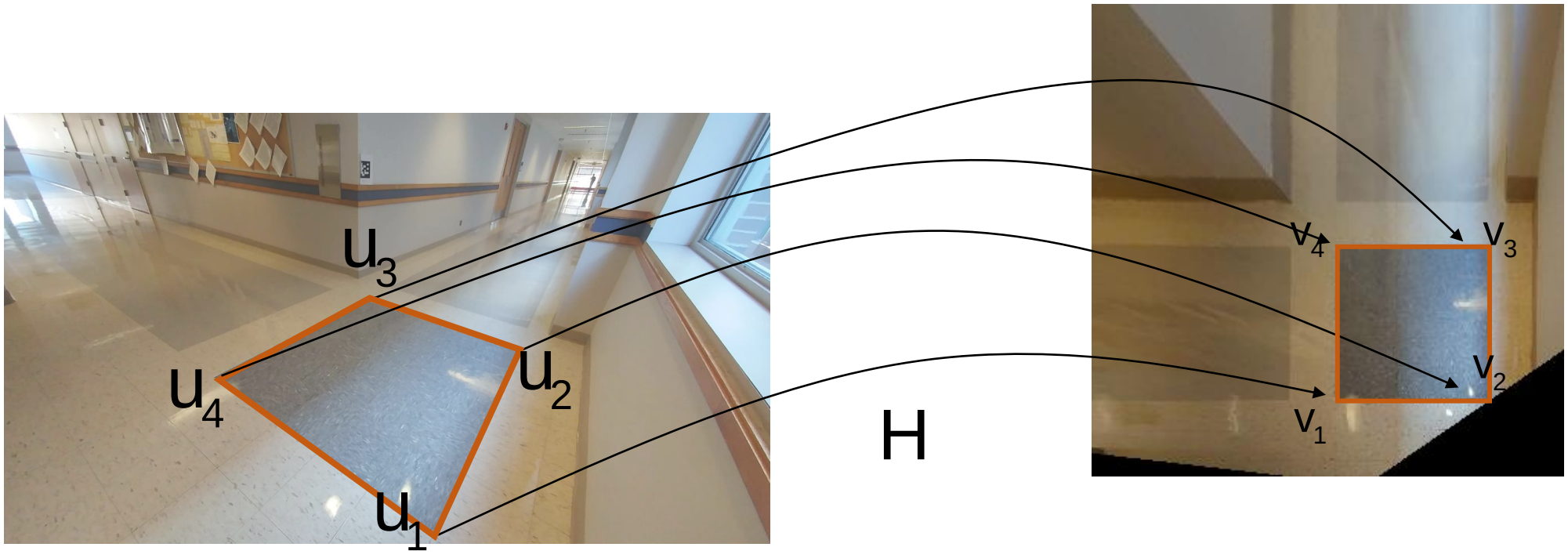
$$\lambda \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}$$

```
im_warped(:,:,1) = reshape(interp2(im(:,:,1), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,2) = reshape(interp2(im(:,:,2), u_x(:), u_y(:)), [h, w]);
im_warped(:,:,3) = reshape(interp2(im(:,:,3), u_x(:), u_y(:)), [h, w]);

im_warped = uint8(im_warped);
```

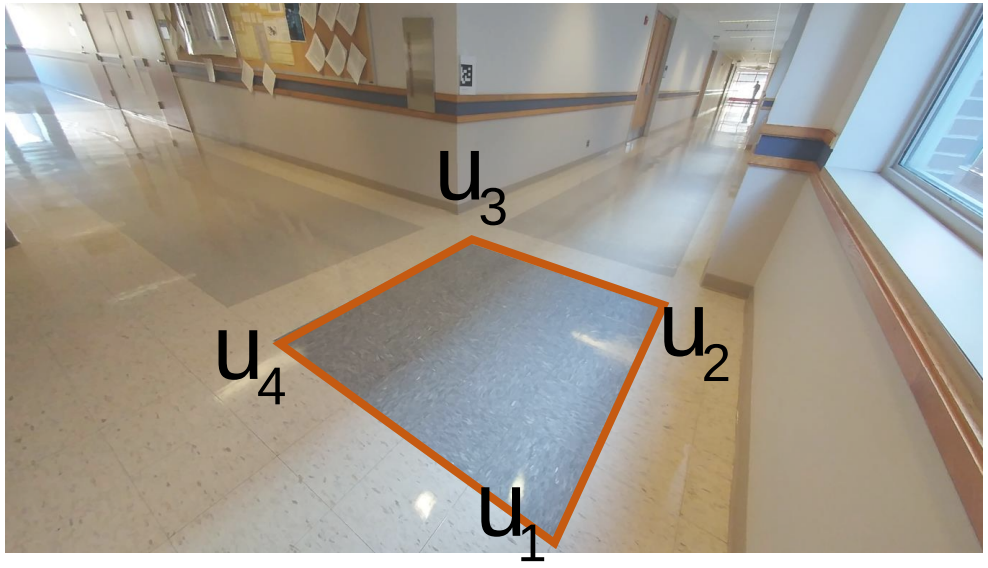
Cf) ImageWarpingEuclidean.m

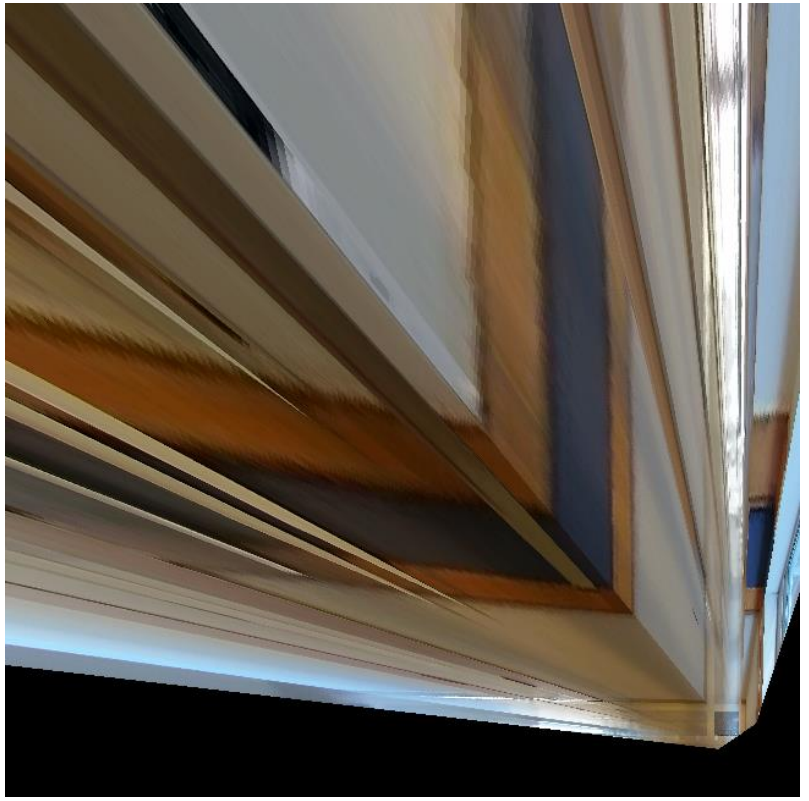
```
u_x = H(1,1)*v_x + H(1,2)*v_y + H(1,3);
u_y = H(2,1)*v_x + H(2,2)*v_y + H(2,3);
```

Fun with Homography



Fun with Homography





Video 5.6
Jianbo Shi

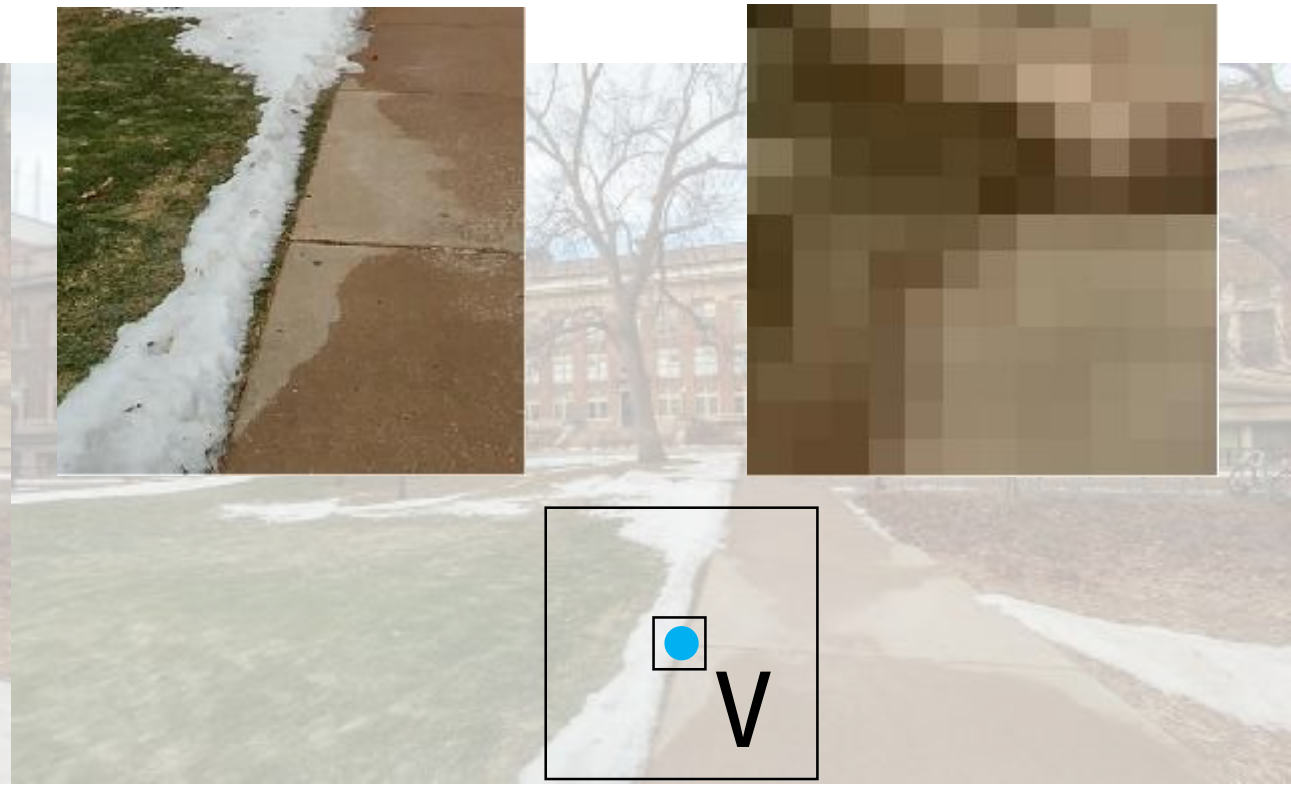
Feature Matching


u
v

Local Patch

Local Patch (Orientation)

Local Patch (Scale)



Local Visual Descriptor

Desired properties:

- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.

Local Visual Descriptor

Desired properties:

- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.
- Orientation aware

Local Scale Invariant Feature Transform (SIFT)

SIFT automatically finds the optimal scale of feature point and its orientation.

Desired properties:

- Repeatability: the same point is repeatedly detected.
- Discriminativity: the point is unique.
- Orientation aware

Local Scale Invariant Feature Transform (SIFT)

Local visual descriptor

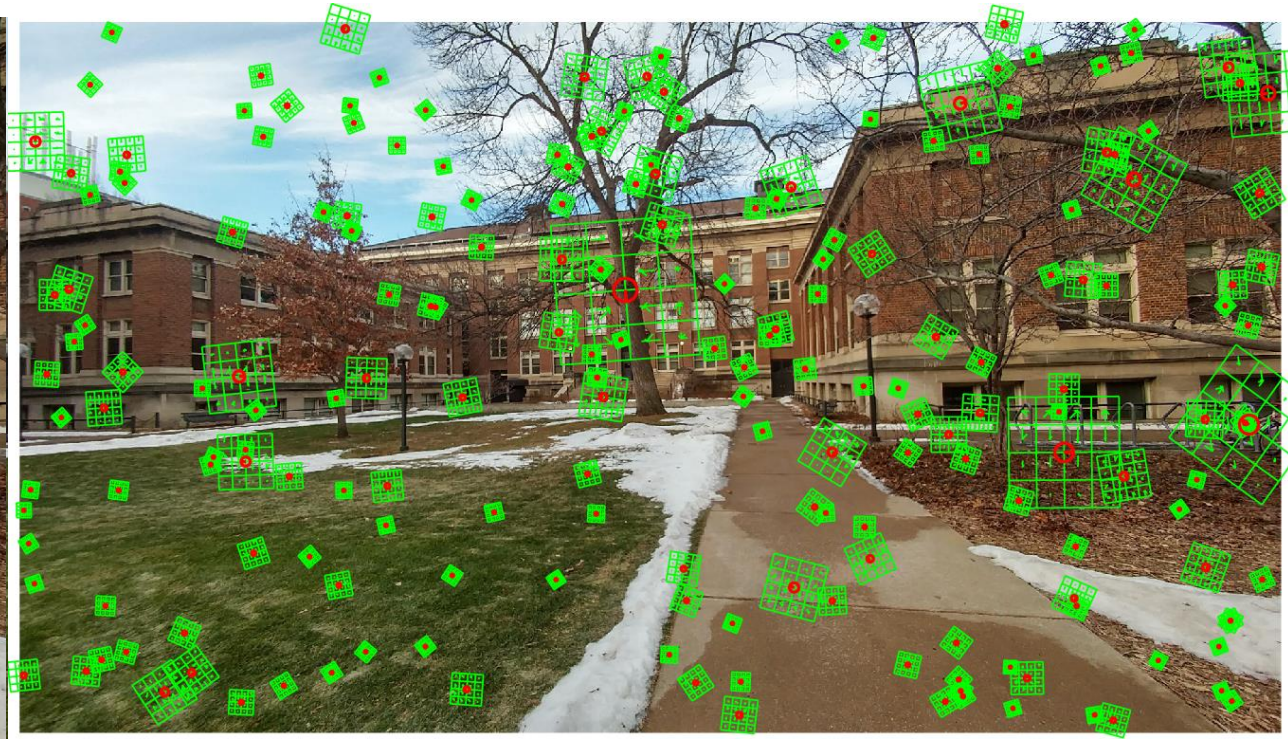
Local Scale Invariant Feature Transform (SIFT)

Local visual descriptor

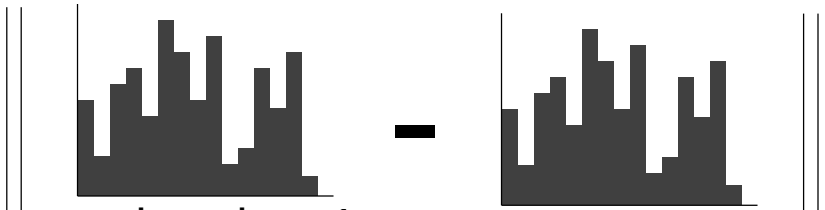
Local Scale Invariant Feature Transform (SIFT)

$$\left\| \text{descriptor1} - \text{descriptor2} \right\| = 0$$

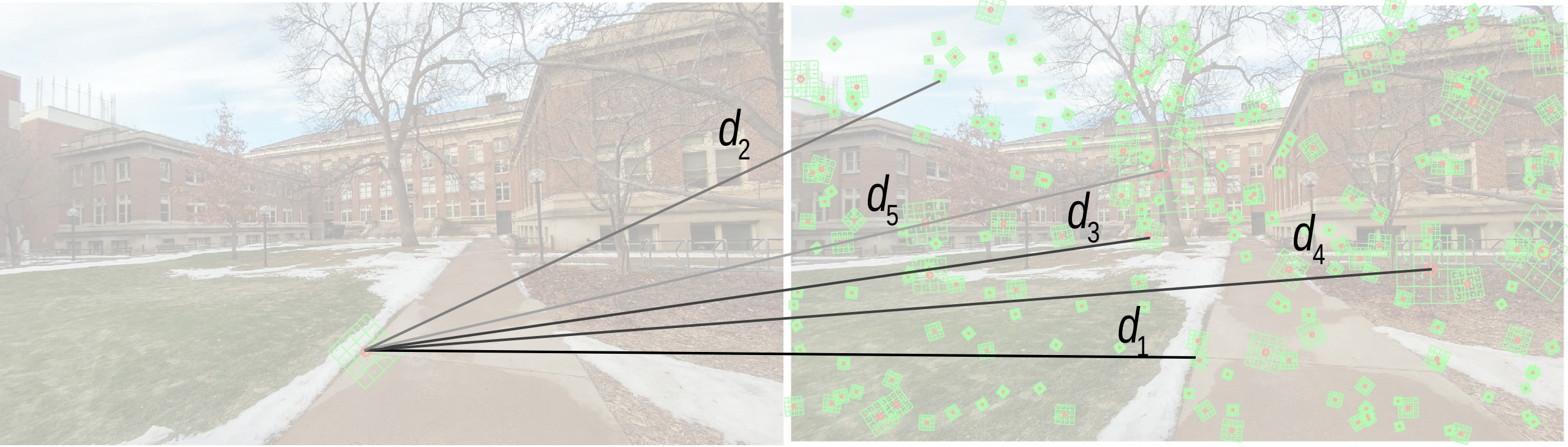
Local Scale Invariant Feature Transform (SIFT)



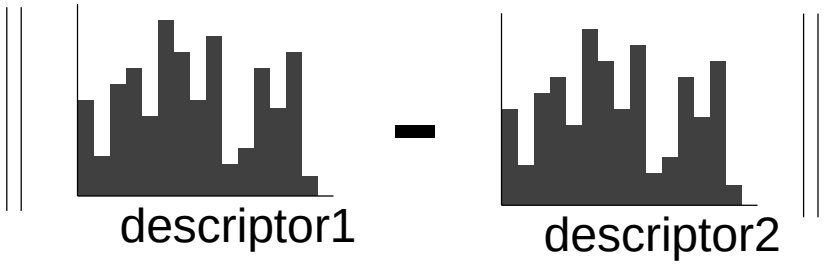
Feature match candidates



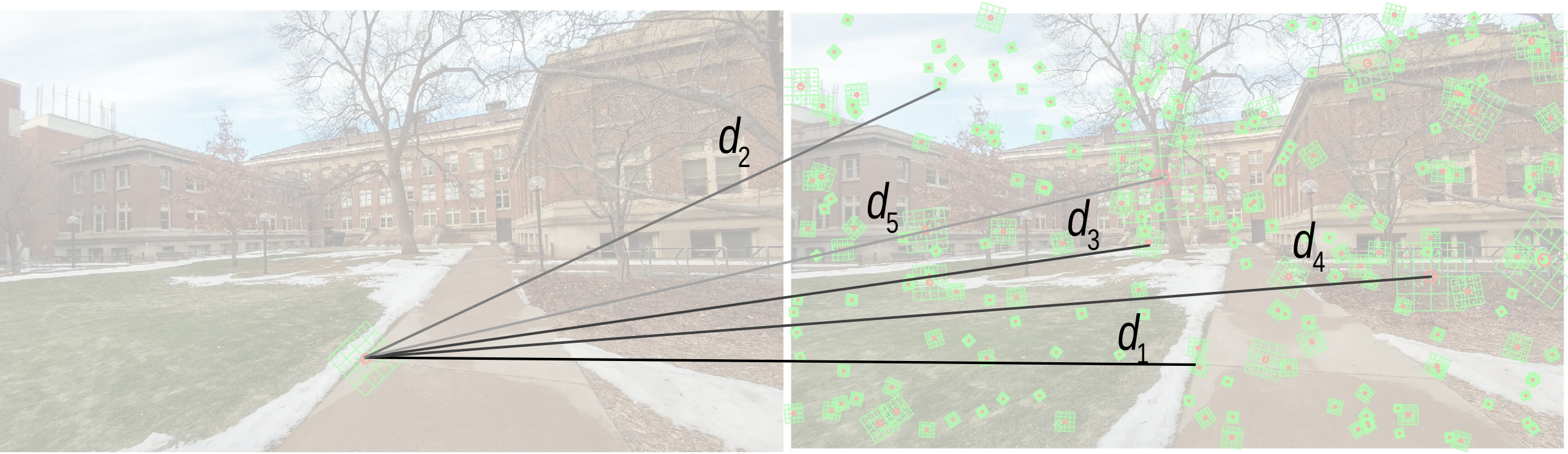
Nearest Neighbor Search



Feature match candidates



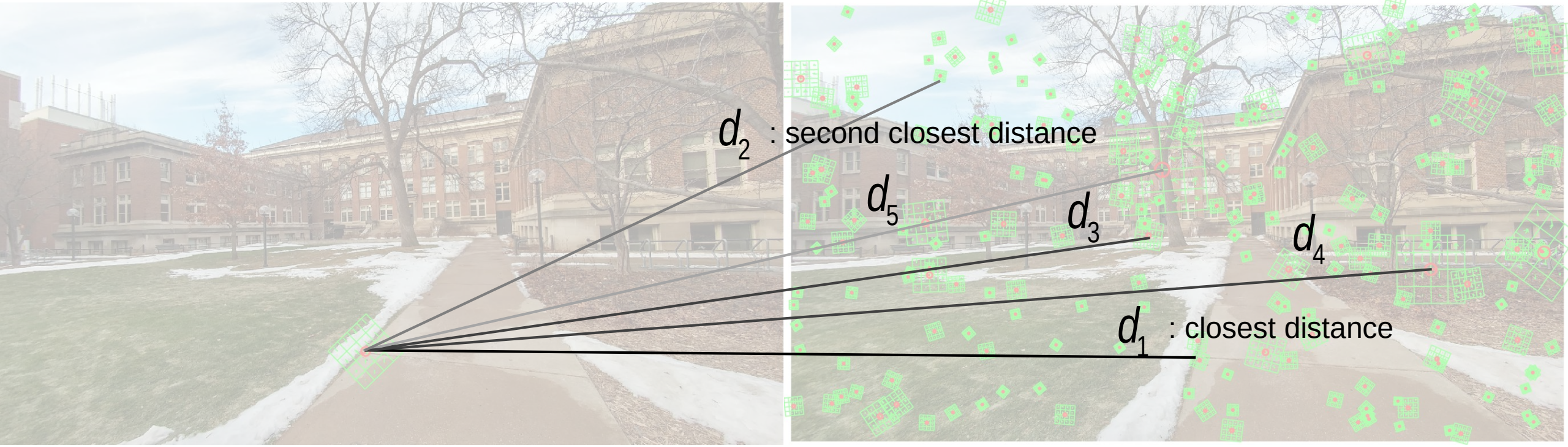
Nearest Neighbor Search



Feature match candidates

Discriminativity: how is the feature point unique?

Nearest Neighbor Search w/ Ratio Test

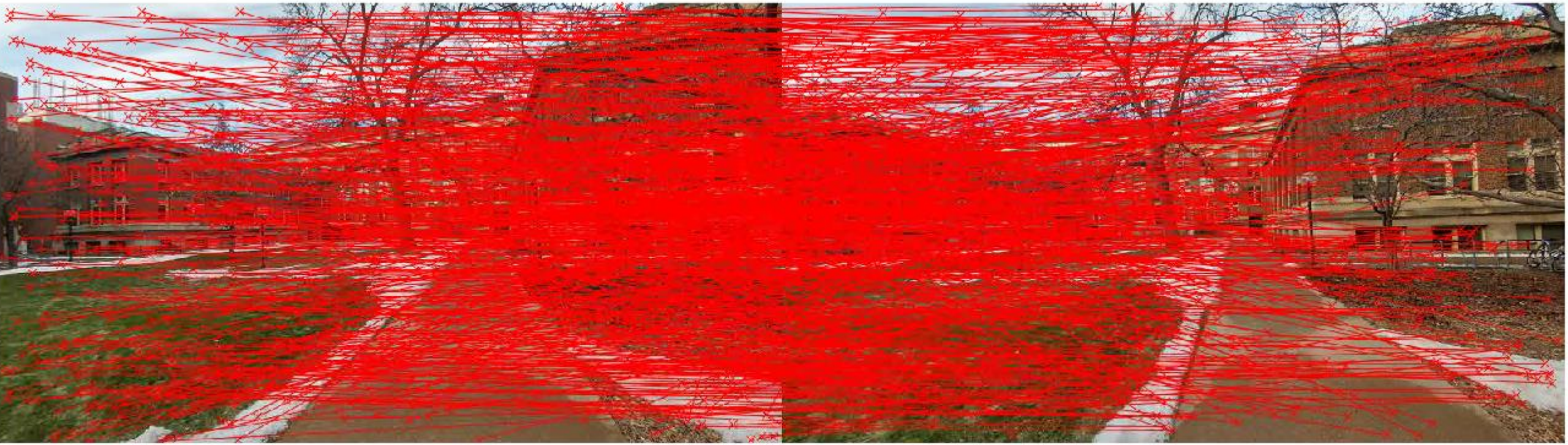


Feature match candidates

Discriminativity: how is the feature point unique?

$$\frac{d_1}{d_2} < 0.7$$

Nearest Neighbor Search w/o Ratio Test

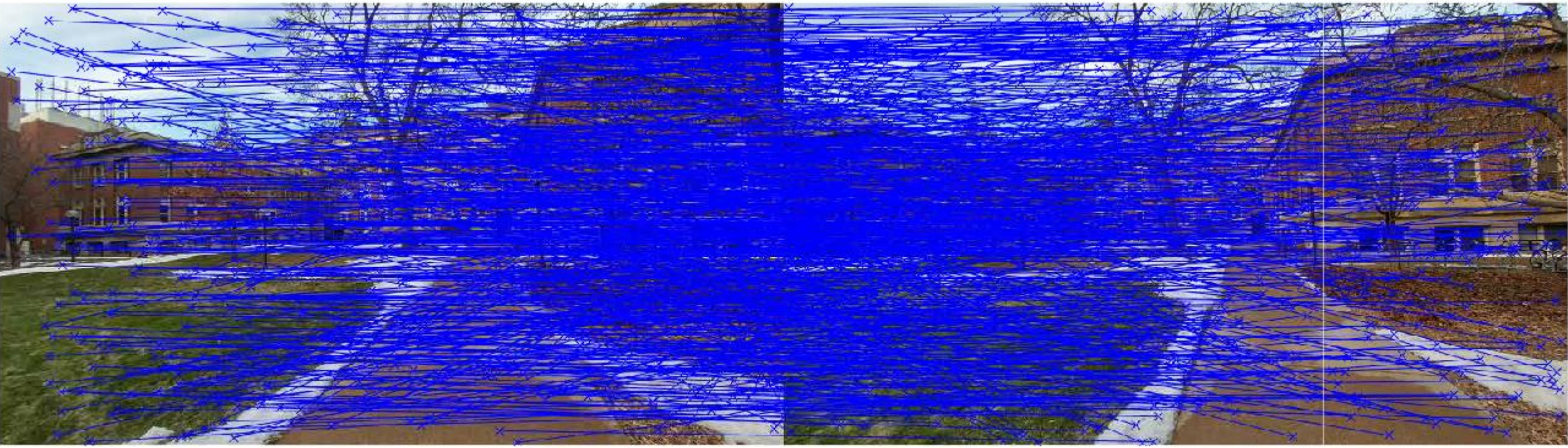


Left image→right image

Nearest Neighbor Search w/ Ratio Test

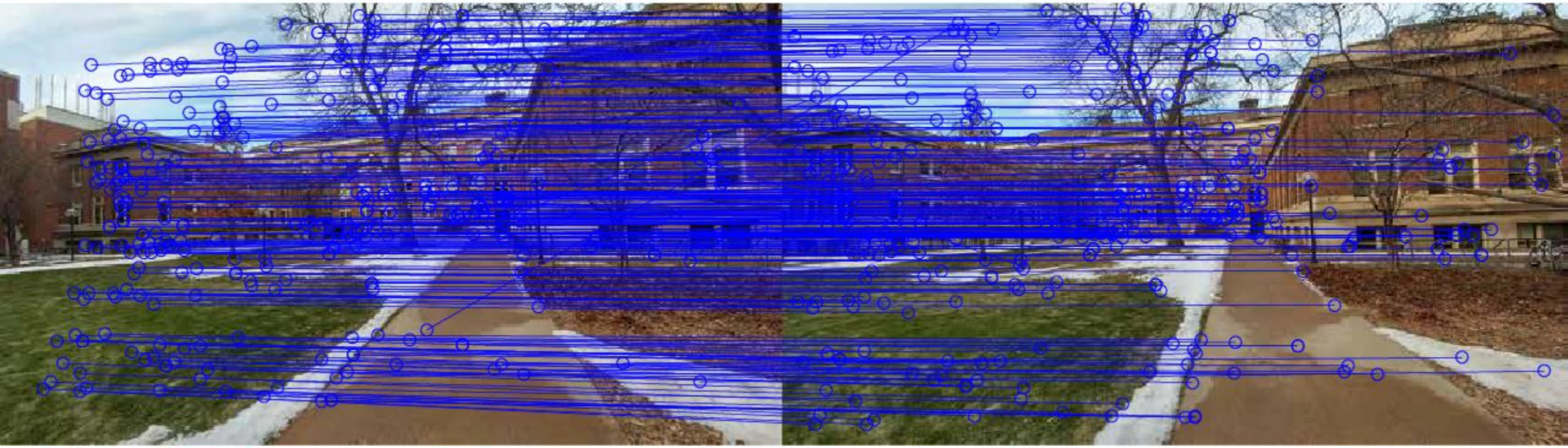
Left image→right image

Nearest Neighbor Search w/o Ratio Test



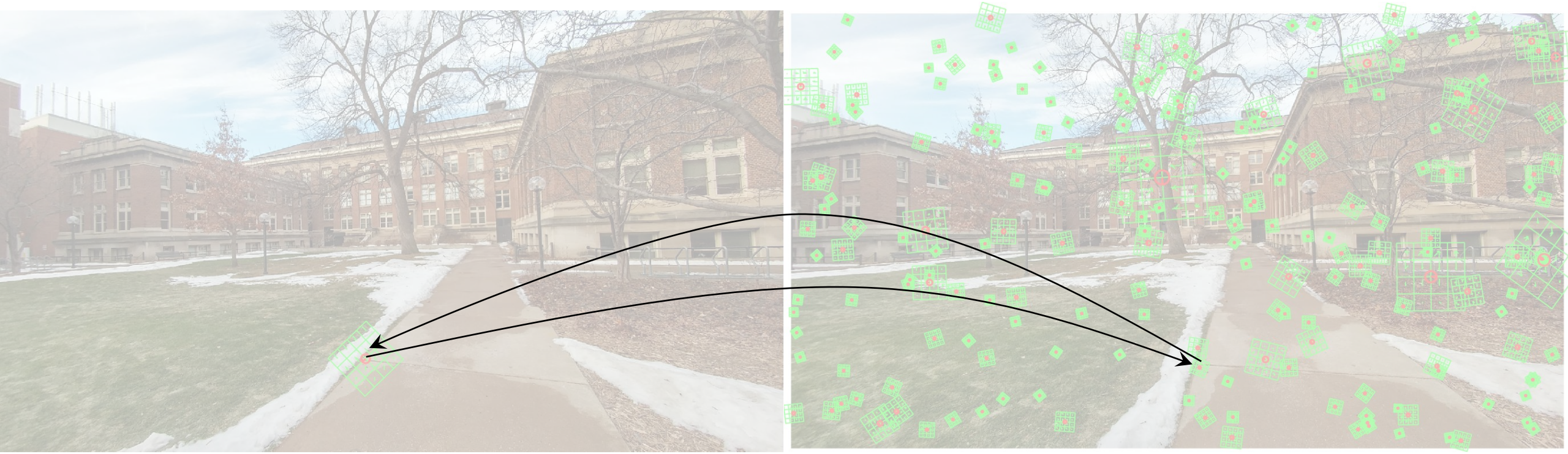
Left image←right image

Nearest Neighbor Search w/ Ratio Test



Left image←right image

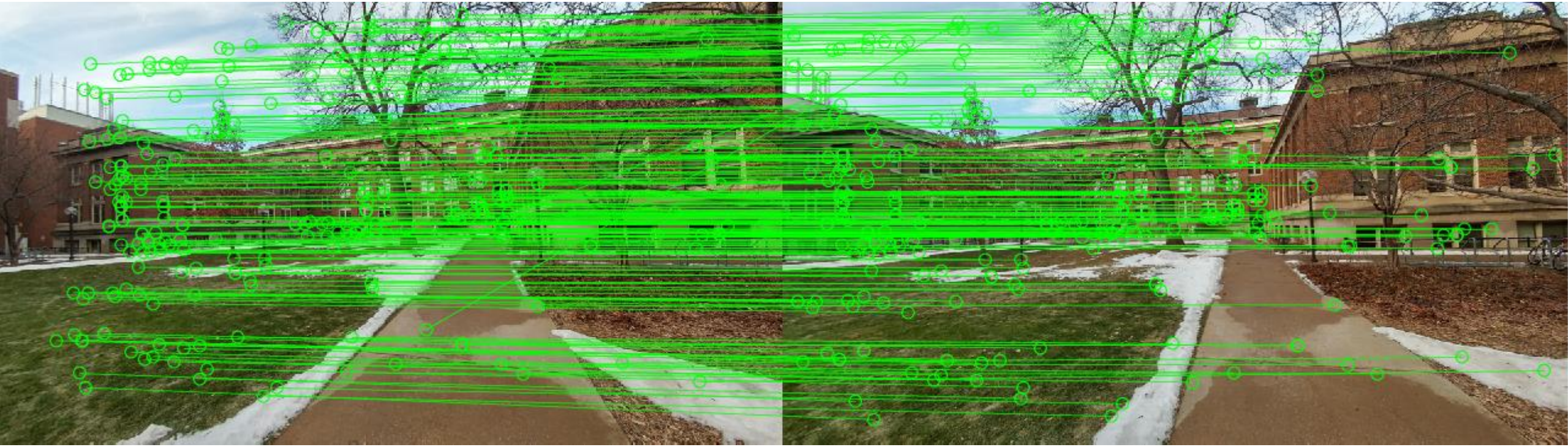
Bi-directional Consistency Check



Feature match candidates

Consistency: would a feature match correspond to each other?

Bi-directional Consistency Check



Video 5.7
Jianbo Shi

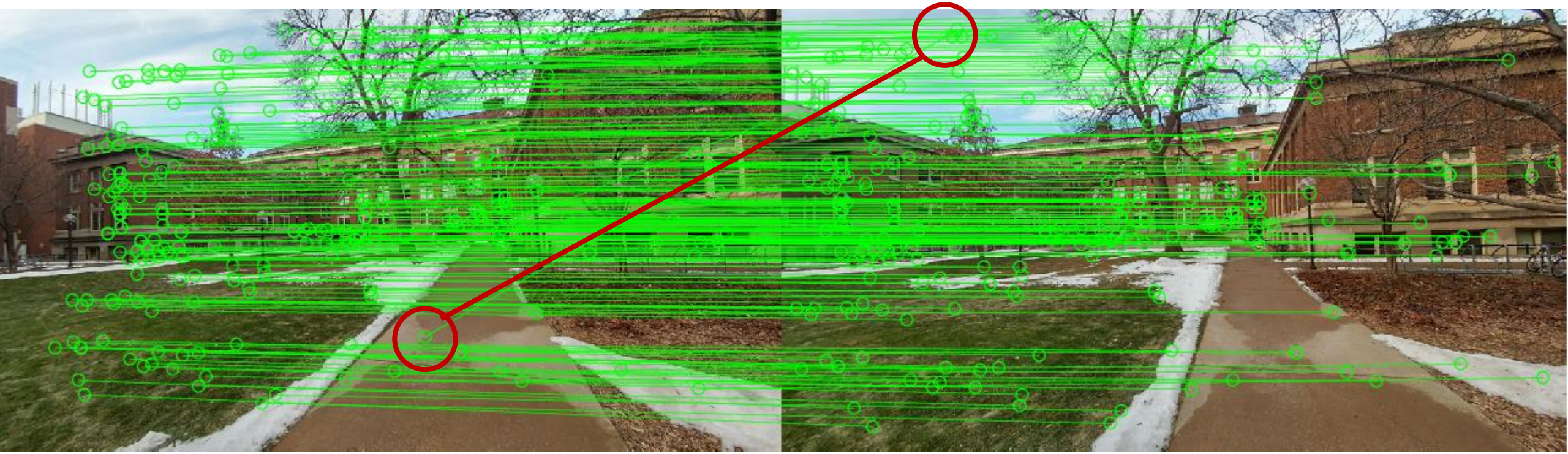
RANSAC: Random Sample Consensus: Linear Least Squares

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

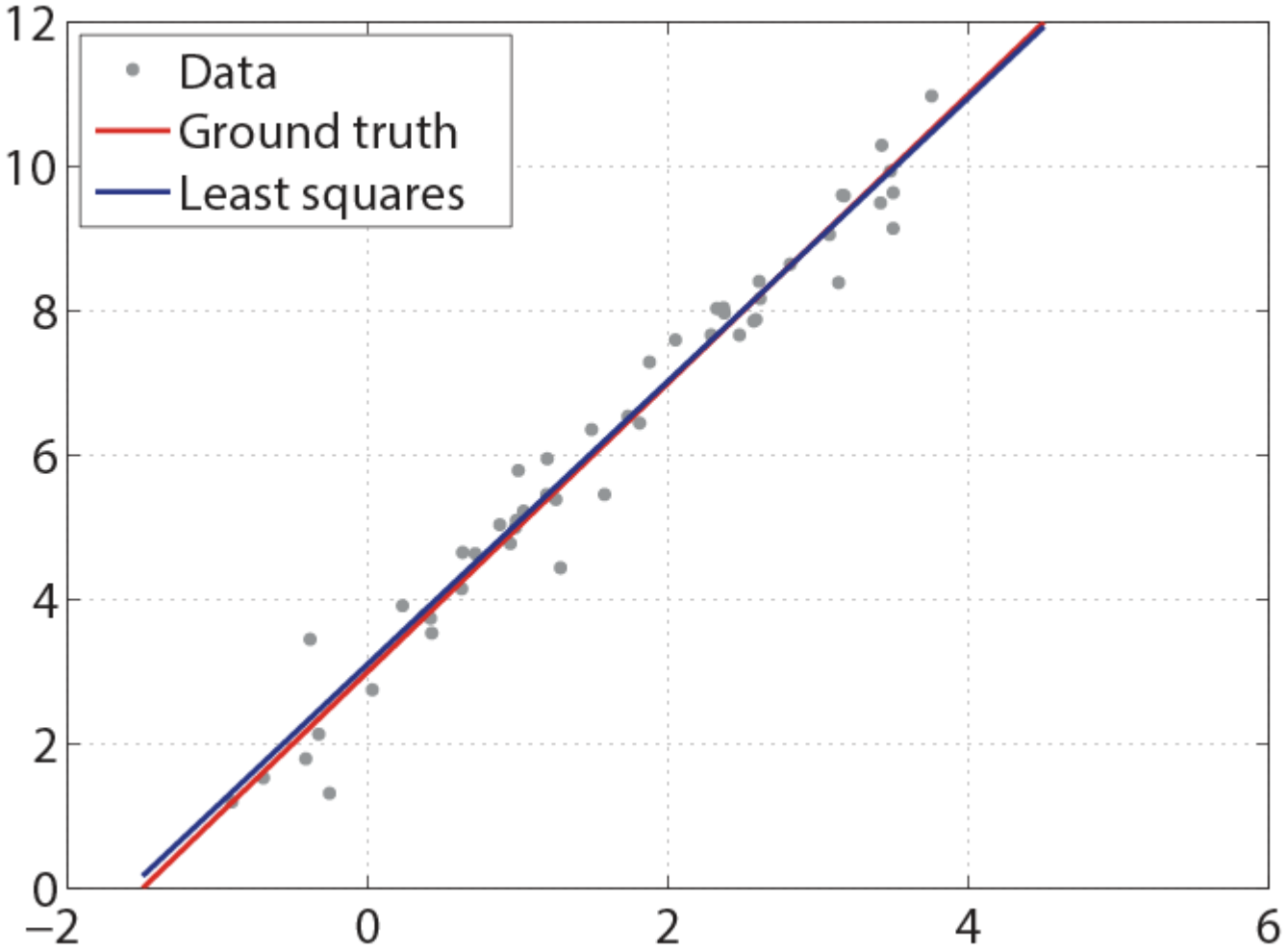
$$\mathbf{x} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}$$

A: 2x9

Outlier?

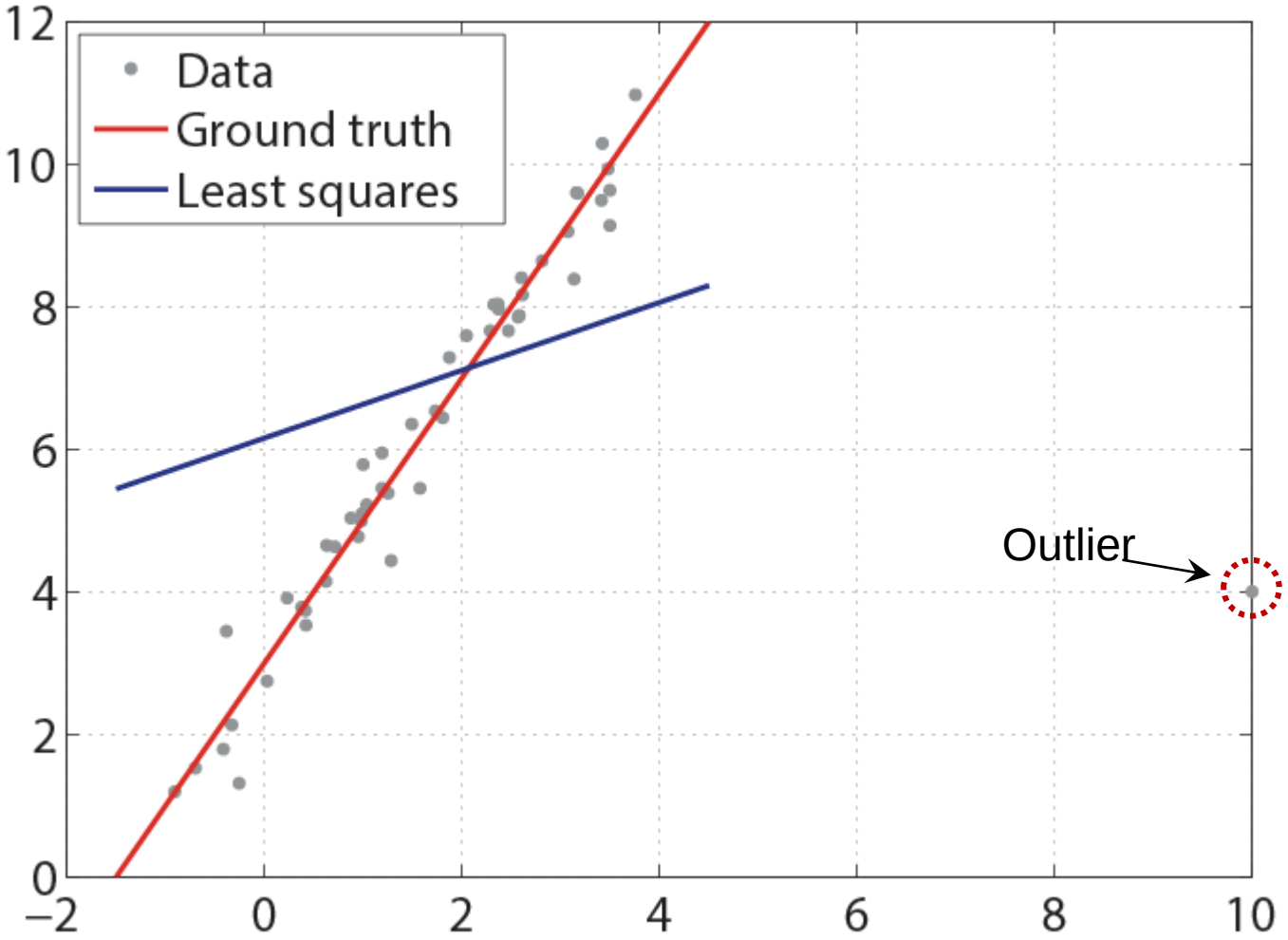


Recall: Line Fitting (Ax=b)

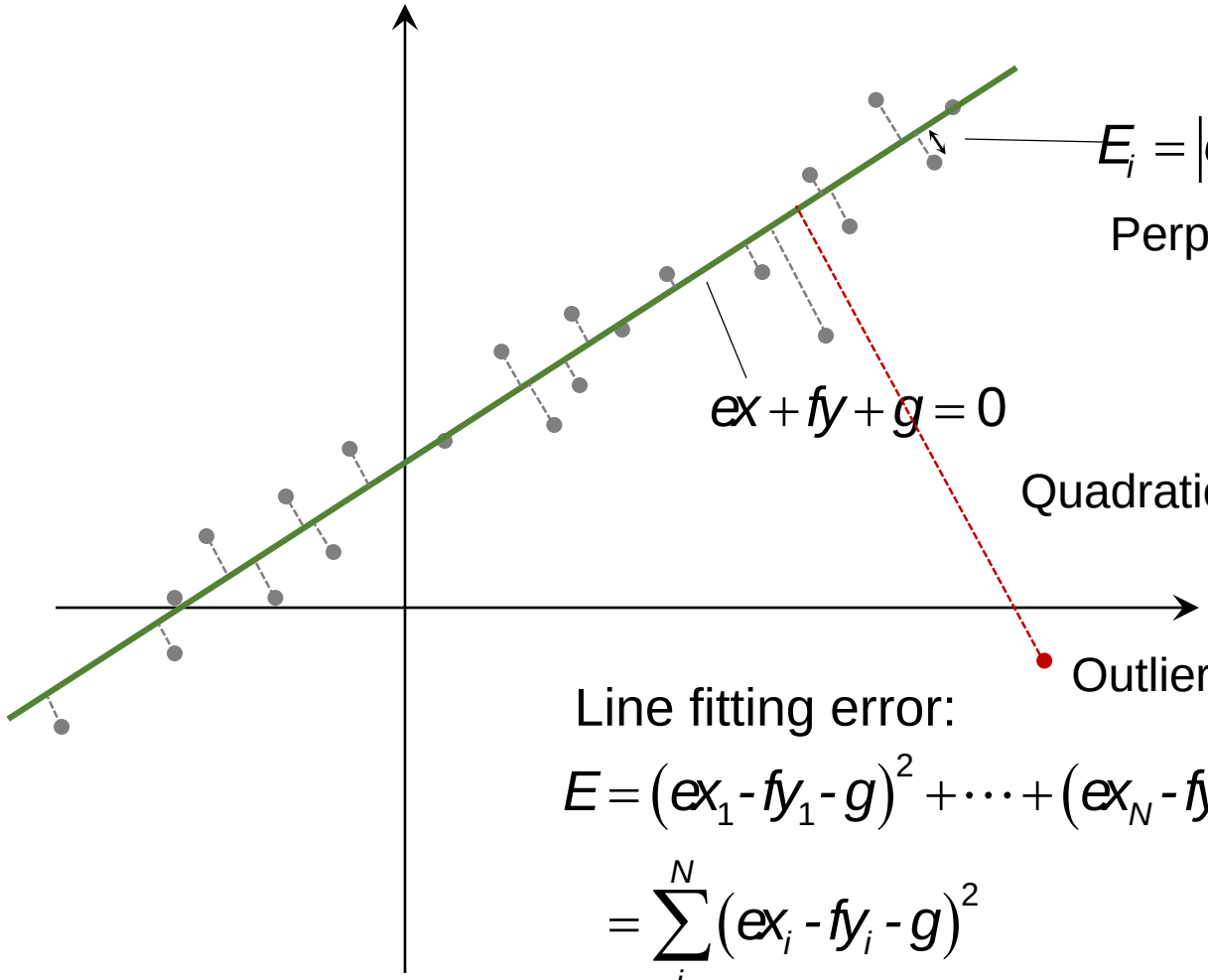


$$\begin{bmatrix} u_x & u_y & 1 & & & & -u_x v_x & -u_y v_x & -v_x \\ & & & u_x & u_y & 1 & -u_x v_y & -u_y v_y & -v_y \end{bmatrix} \underbrace{\quad A \quad}_{2\times 9} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} b$$

Outlier



$$\begin{bmatrix} u_x & u_y & 1 & 0 & 0 & 0 & -u_x v_x & -u_y v_x & -v_x \\ 0 & 0 & 0 & u_x & u_y & 1 & -u_x v_y & -u_y v_y & -v_y \end{bmatrix} \underbrace{\quad A \quad}_{2\times 9} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} x = \begin{bmatrix} 0 \\ 0 \end{bmatrix} b$$



$E_i = |ax_i + by_i + c|$

Perpendicular distance

$ax + by + c = 0$

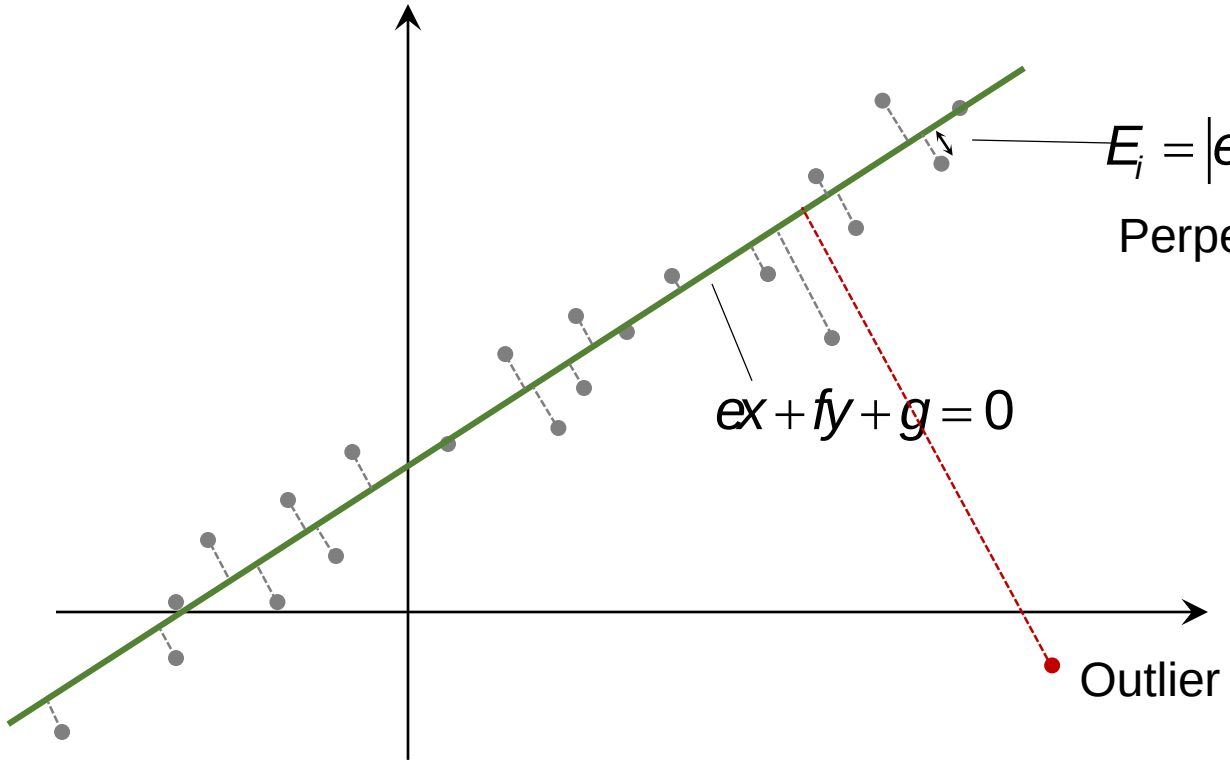
Quadratic magnification of error of outliers

Outlier

Line fitting error:

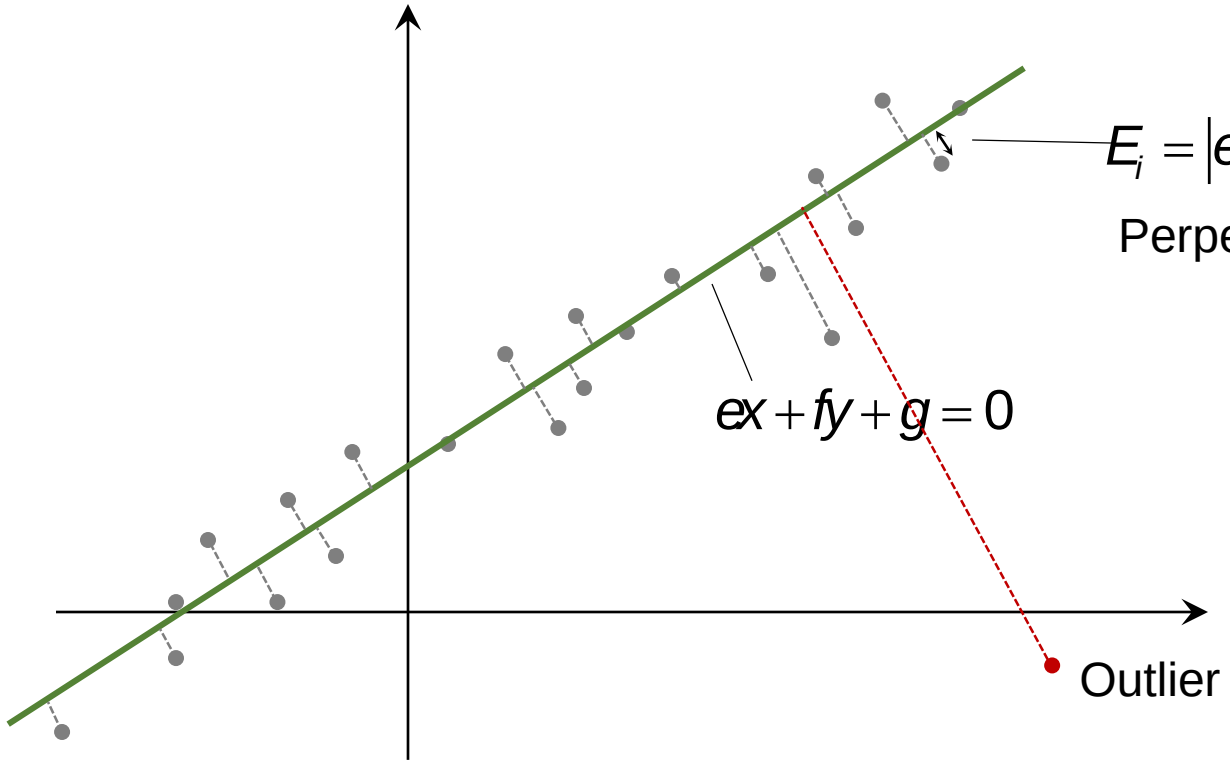
$$E = (ax_1 - by_1 - c)^2 + \cdots + (ax_N - by_N - c)^2$$

$$= \sum_i^N (ax_i - by_i - c)^2$$



Outlier rejection strategy:

To find the best line that explanes the maximum number of points.



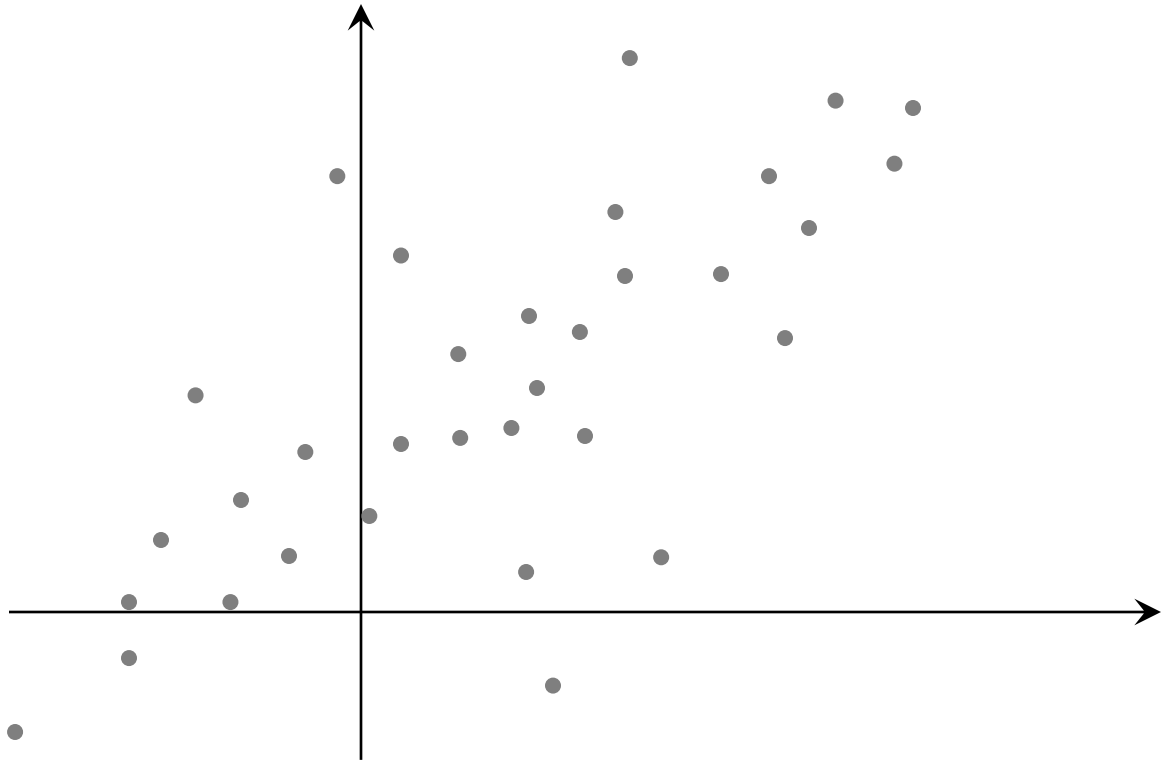
Outlier rejection strategy:

To find the best line that explanes the maximum number of points.

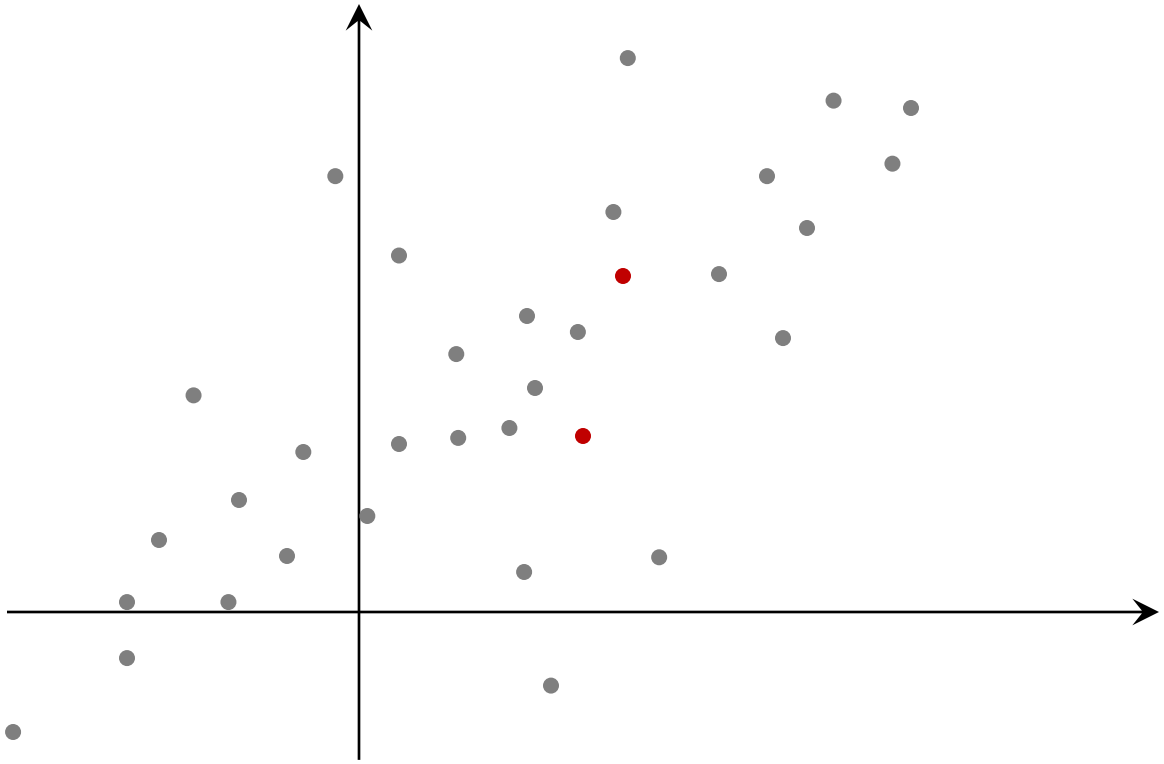
Assumptions:

1. Majority of good samples agree with the underlying model (good apples are same and simple.).

2. Bad samples does not consistently agree with a single model
(all bad apples are different and complicated.).

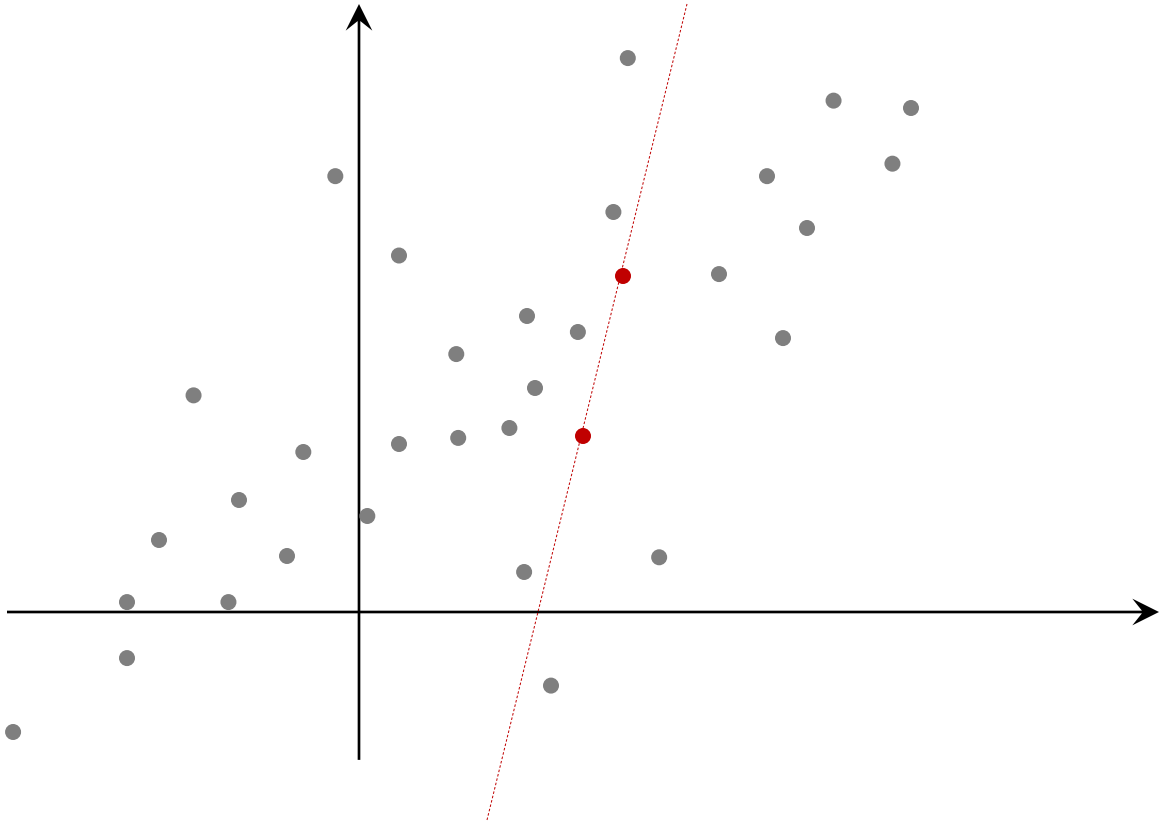


RANSAC: Random Sample Consensus



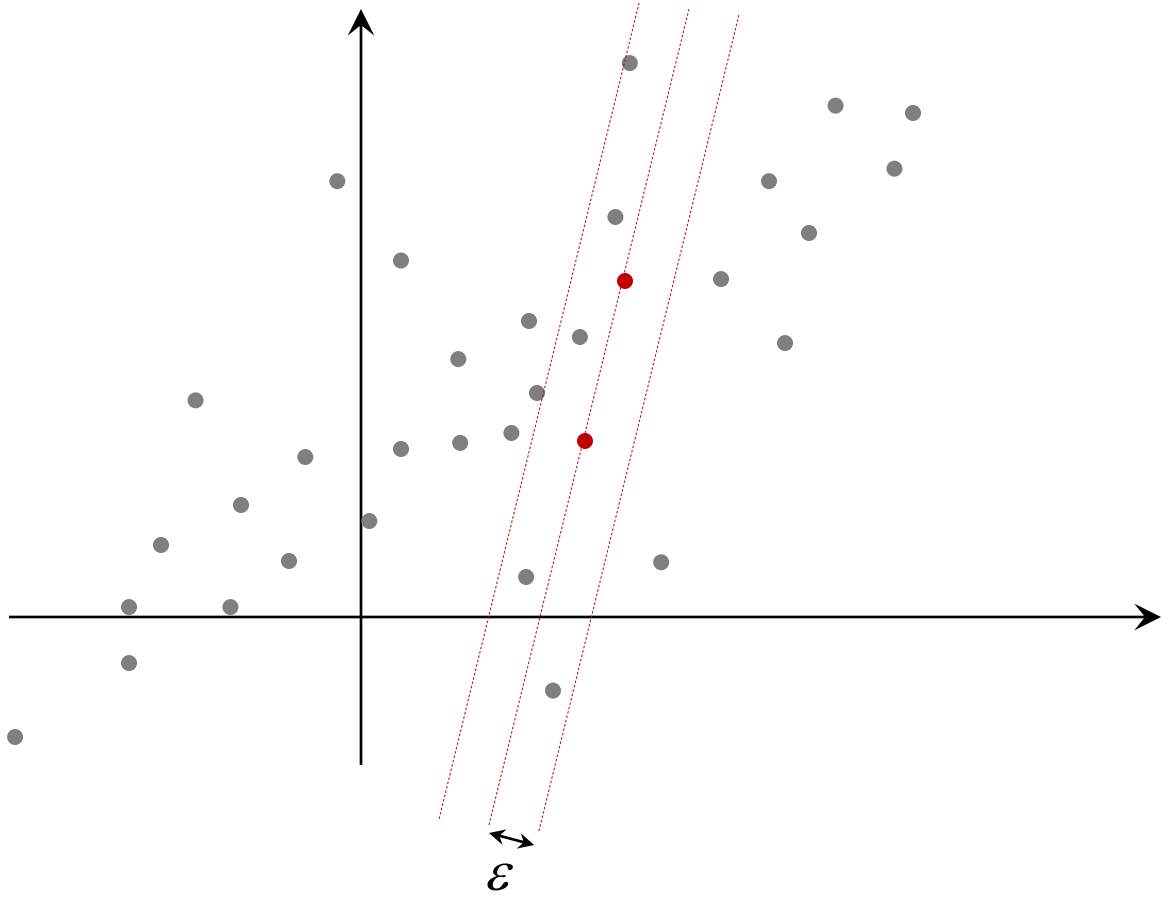
1. Random sampling

RANSAC: Random Sample Consensus



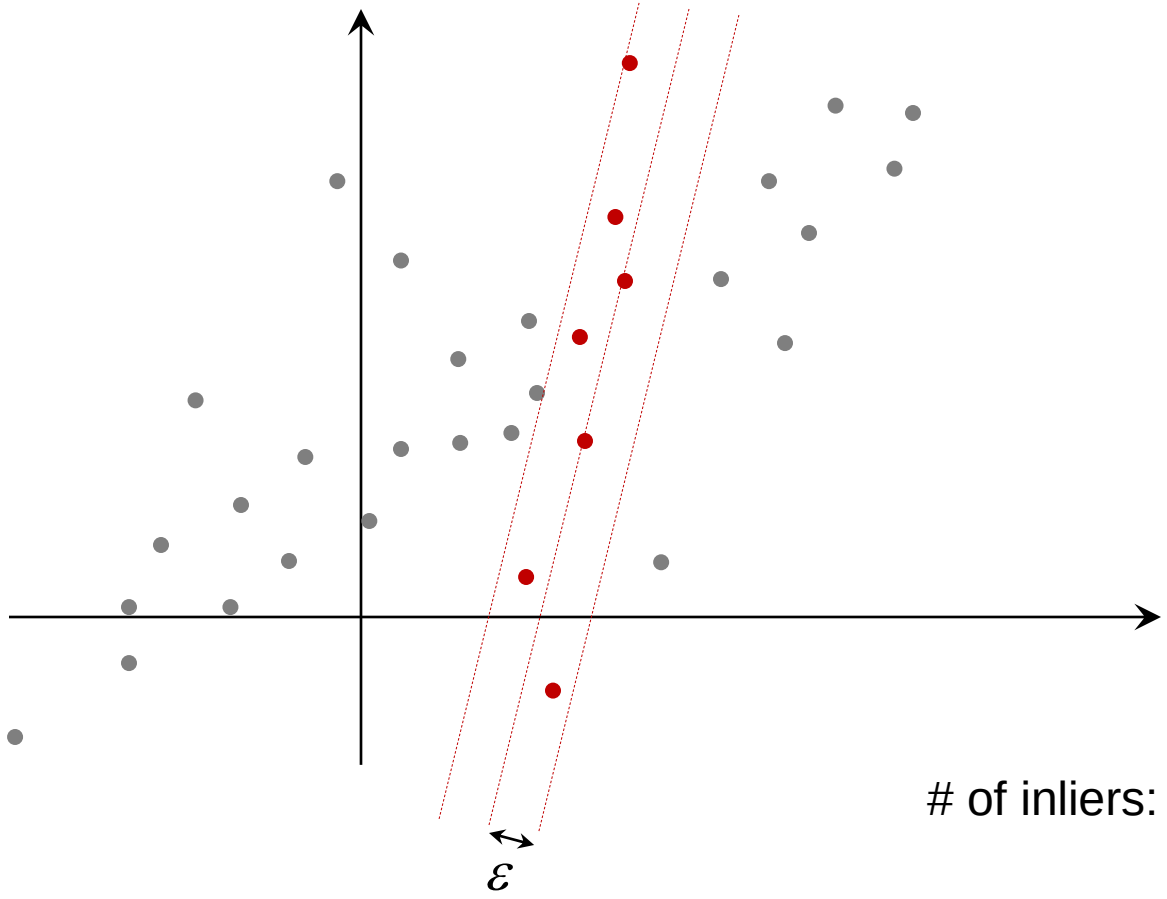
1. Random sampling
2. Model building

RANSAC: Random Sample Consensus



1. Random sampling
2. Model building
3. Thresholding

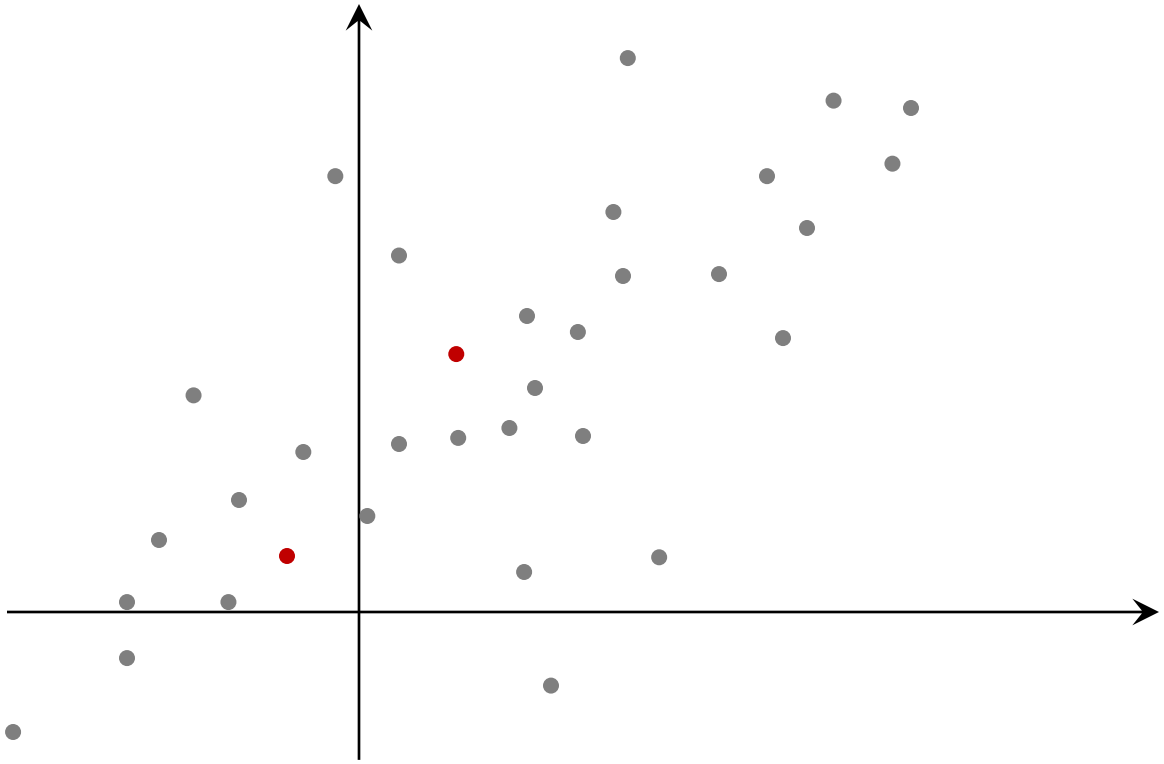
RANSAC: Random Sample Consensus



# of inliers: 7

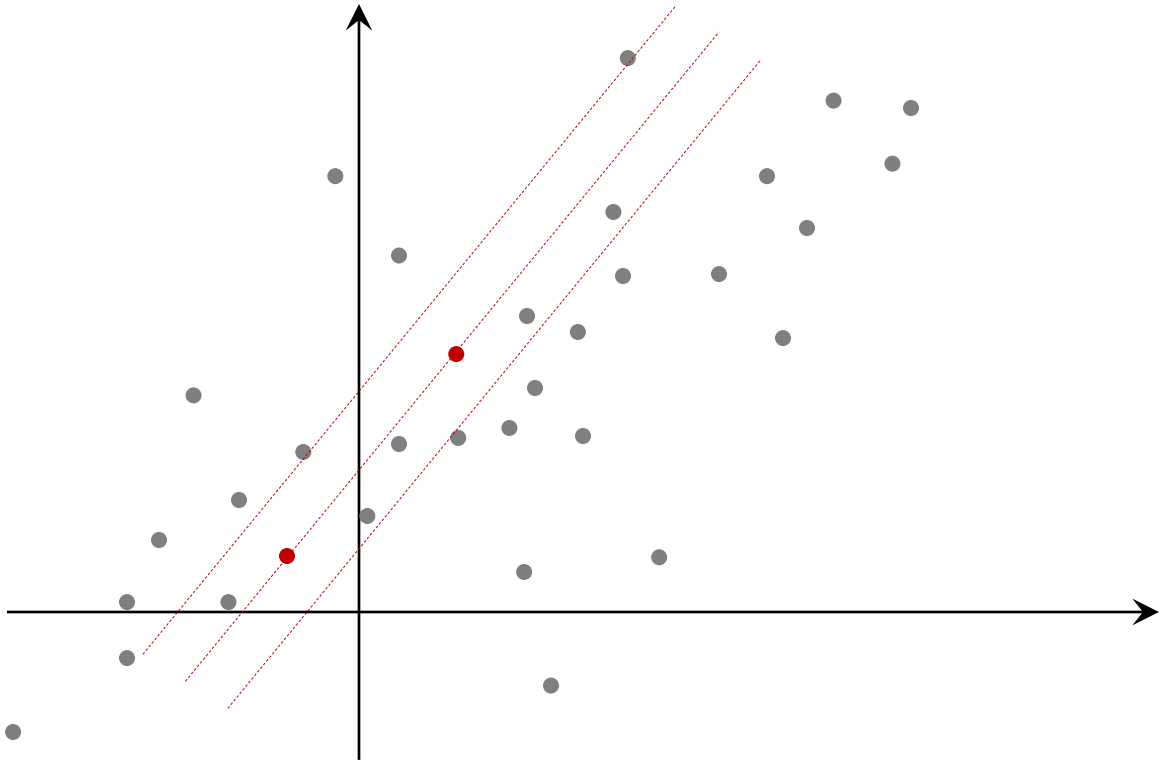
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



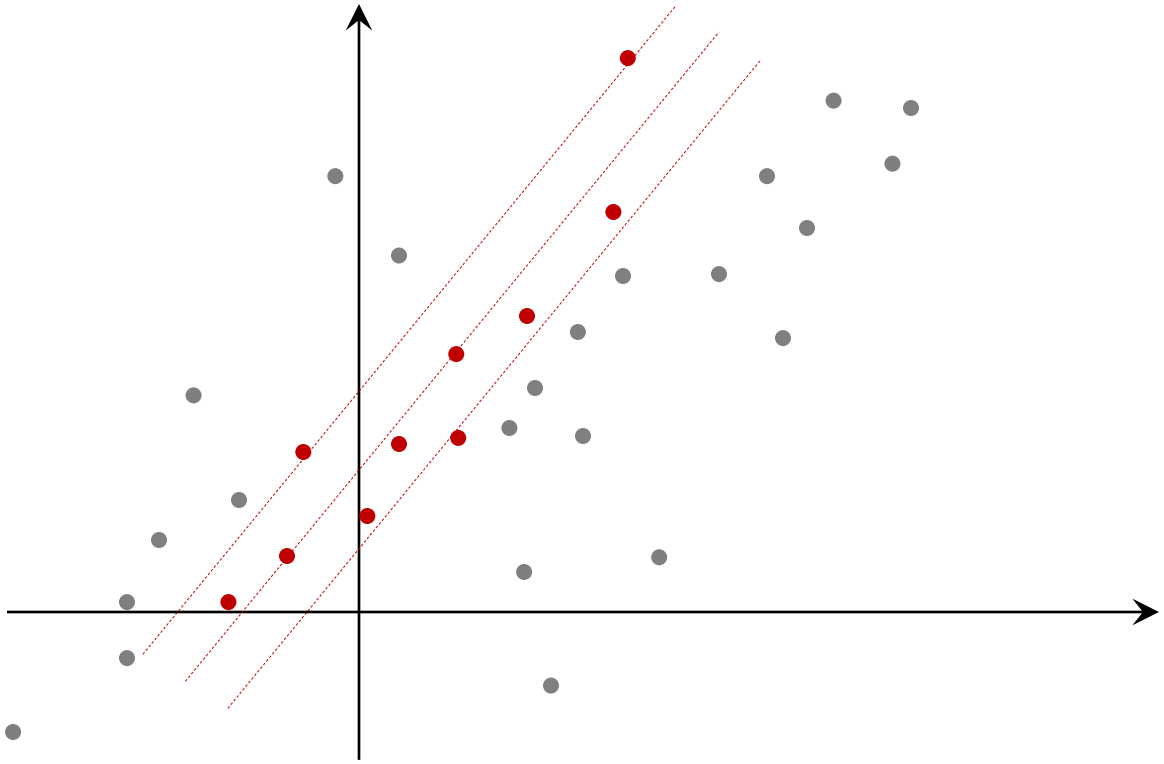
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



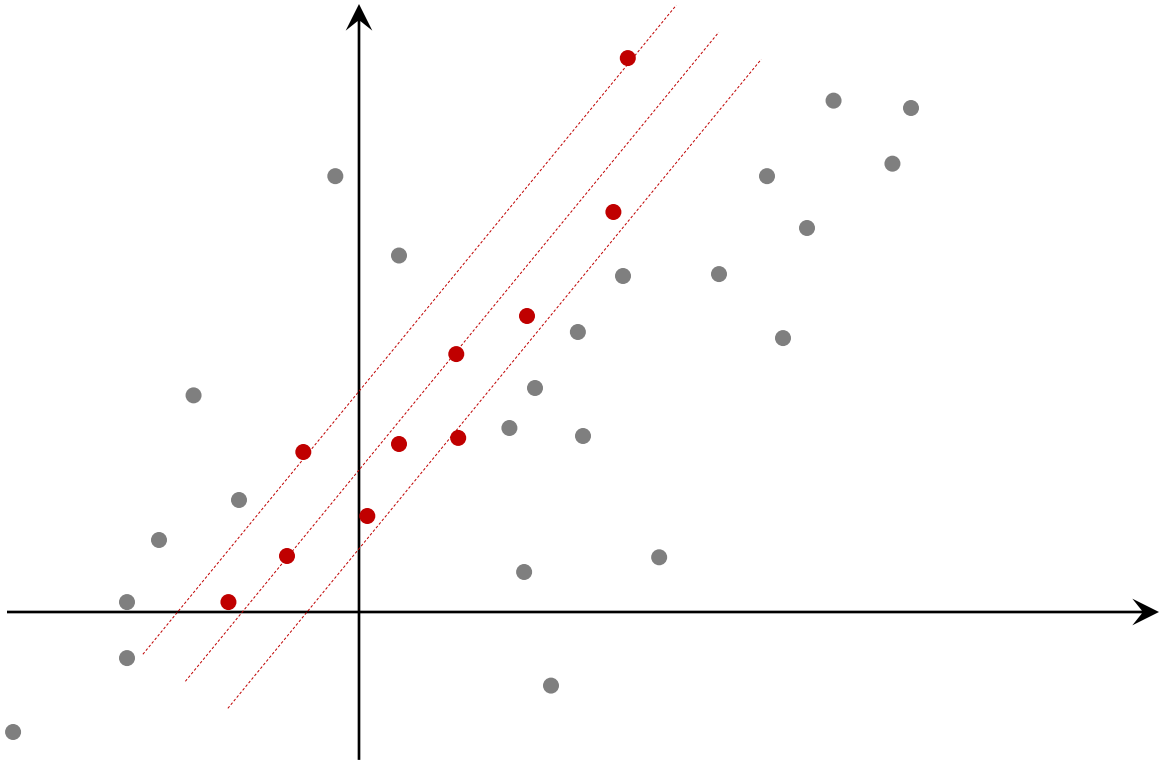
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

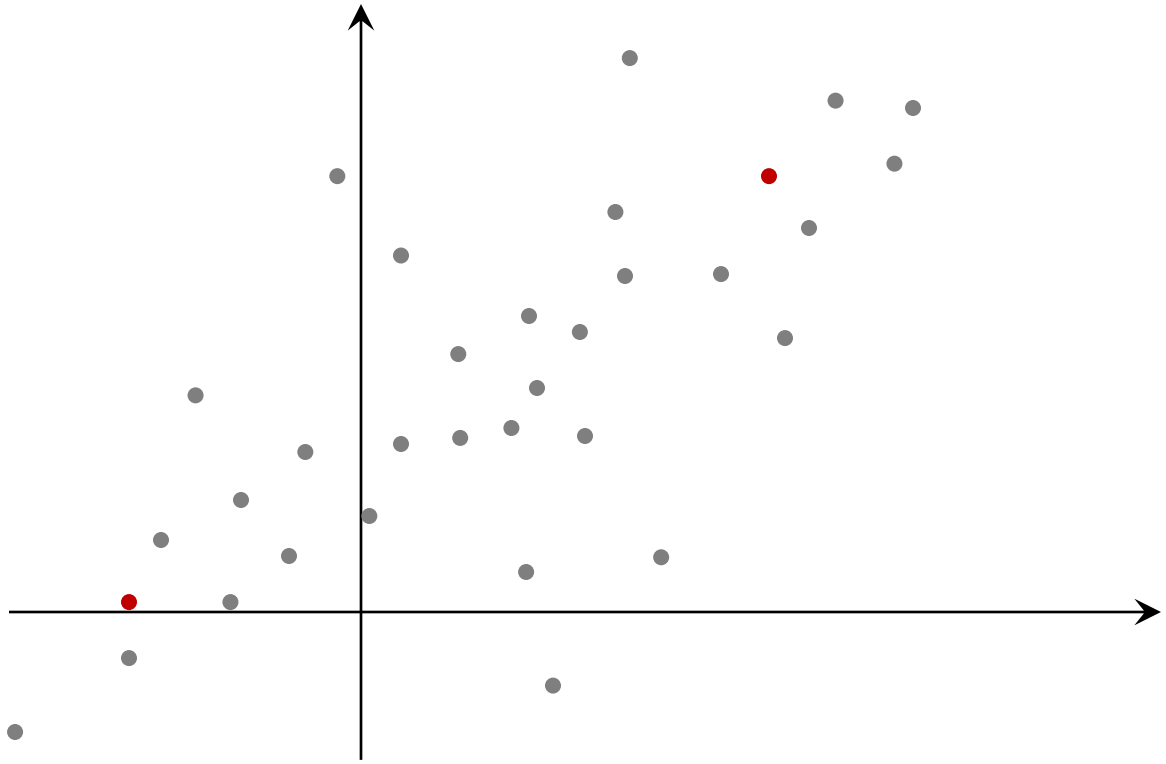
RANSAC: Random Sample Consensus



1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

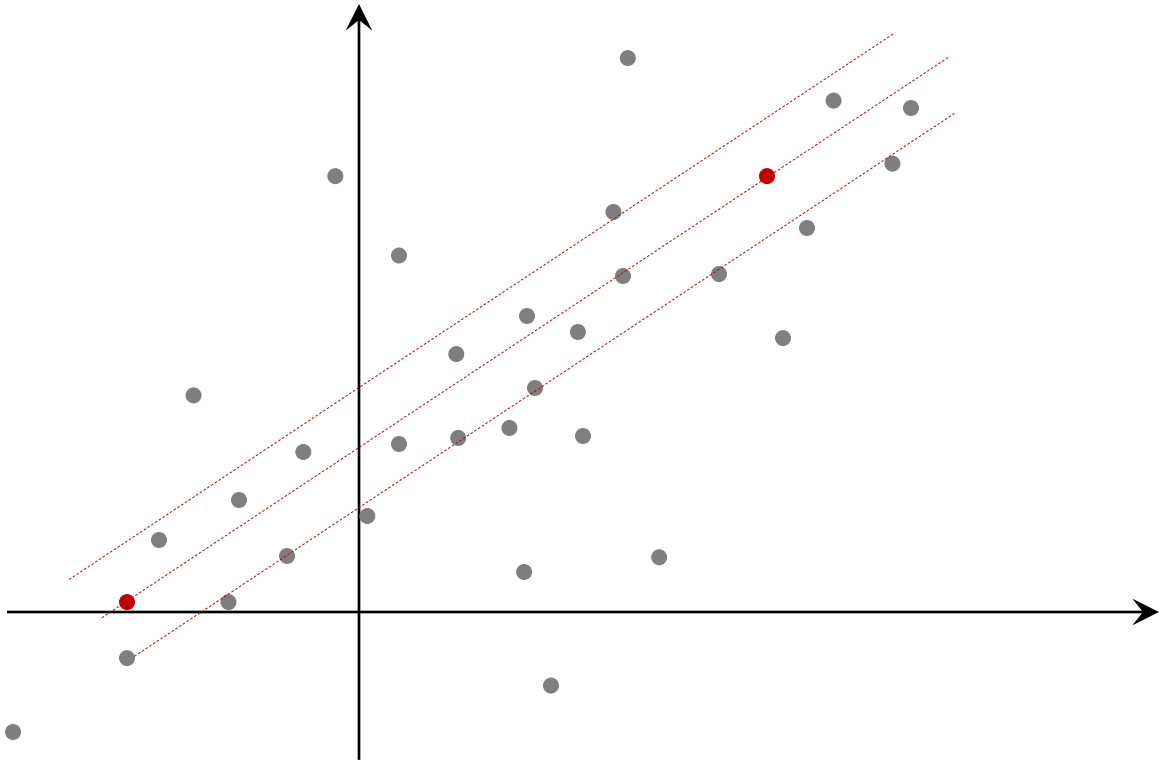
# of inliers: 10

RANSAC: Random Sample Consensus



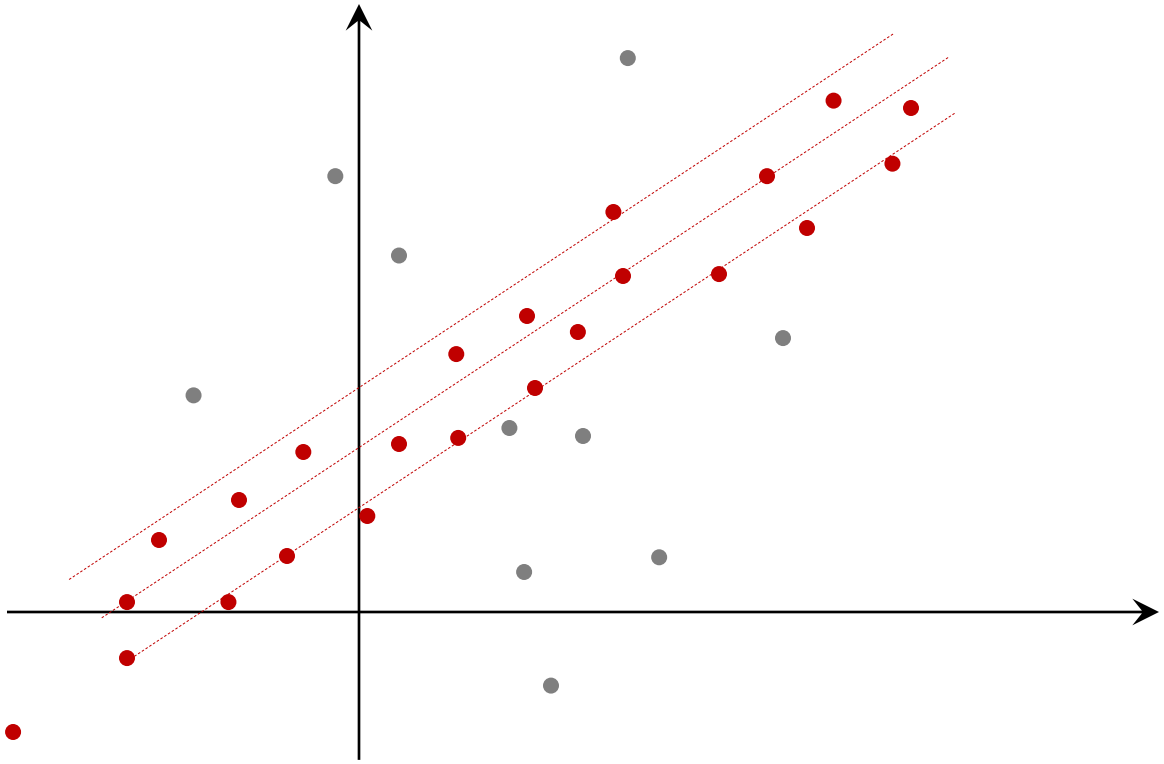
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



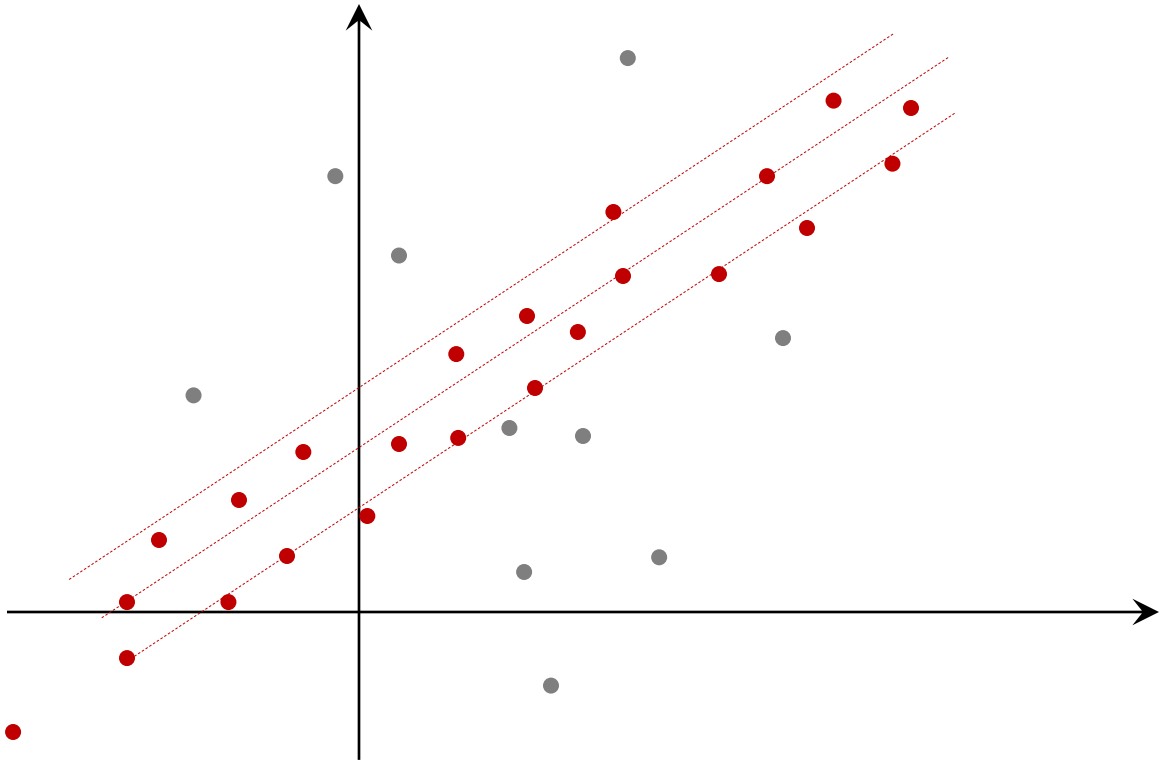
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

RANSAC: Random Sample Consensus



1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

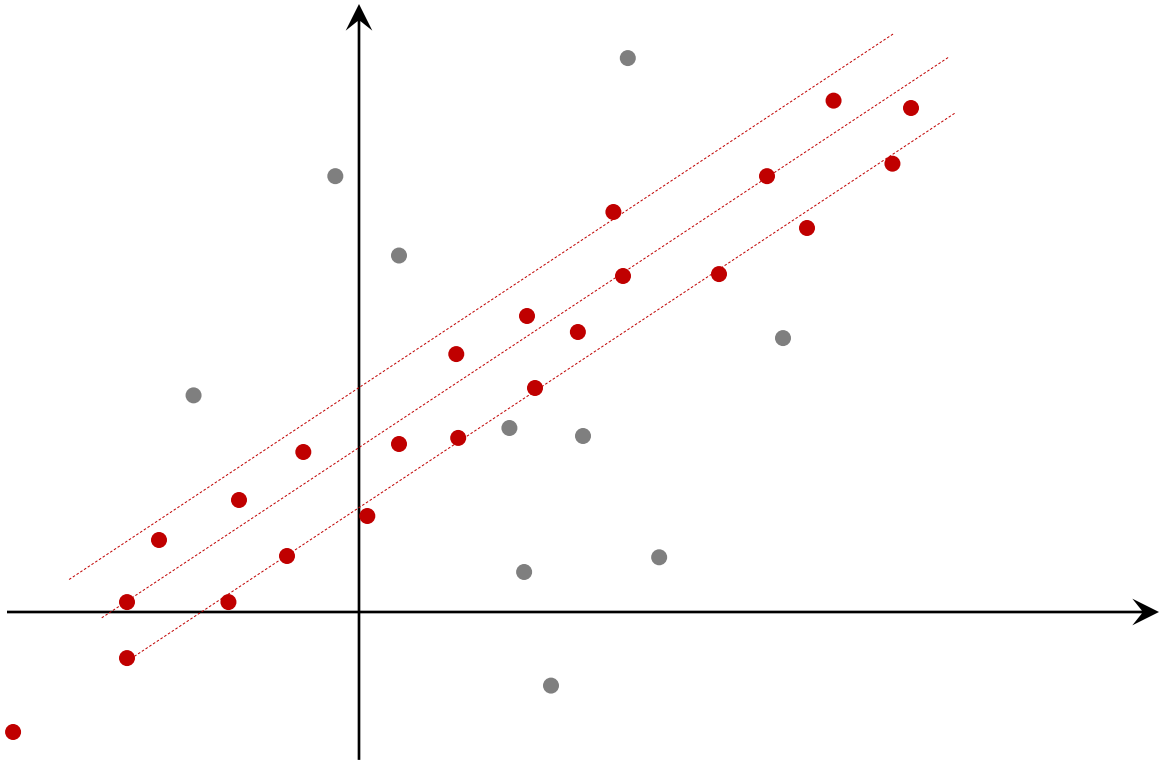
RANSAC: Random Sample Consensus



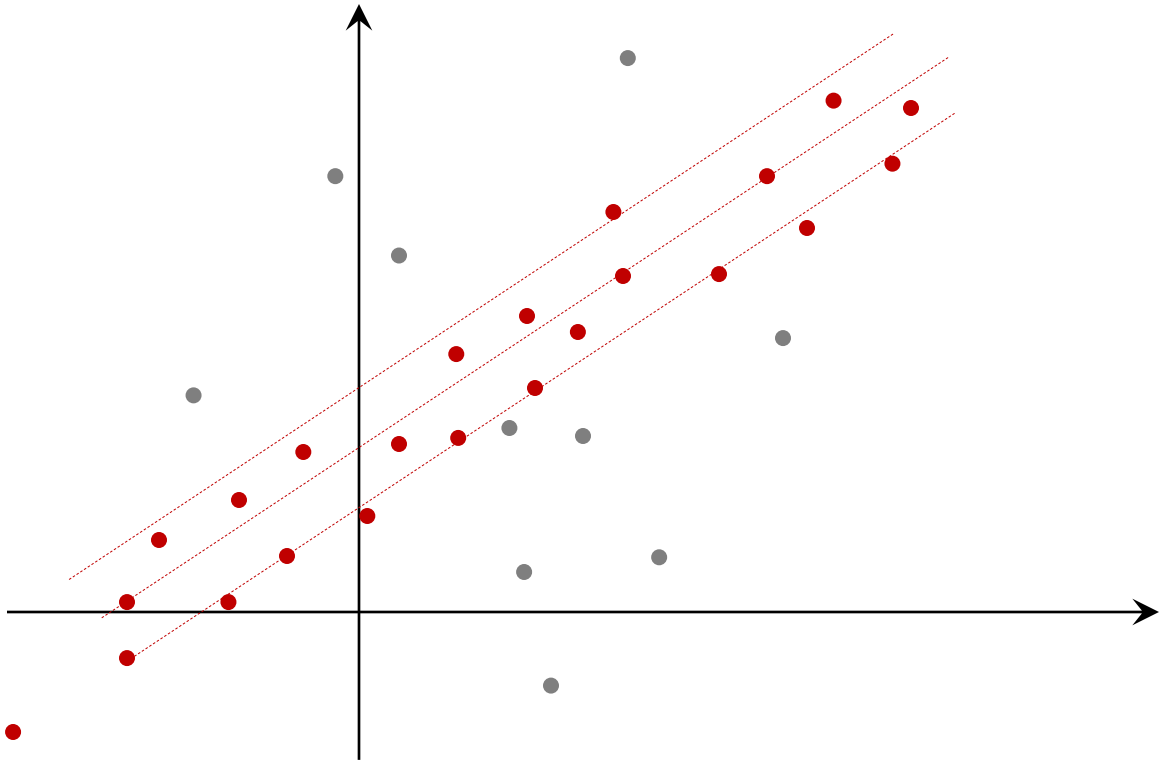
1. Random sampling
2. Model building
3. Thresholding
4. Inlier counting

$$\frac{\text{\# of inliers: 23}}{\text{Maximum number of inliers}}$$

RANSAC: Random Sample Consensus

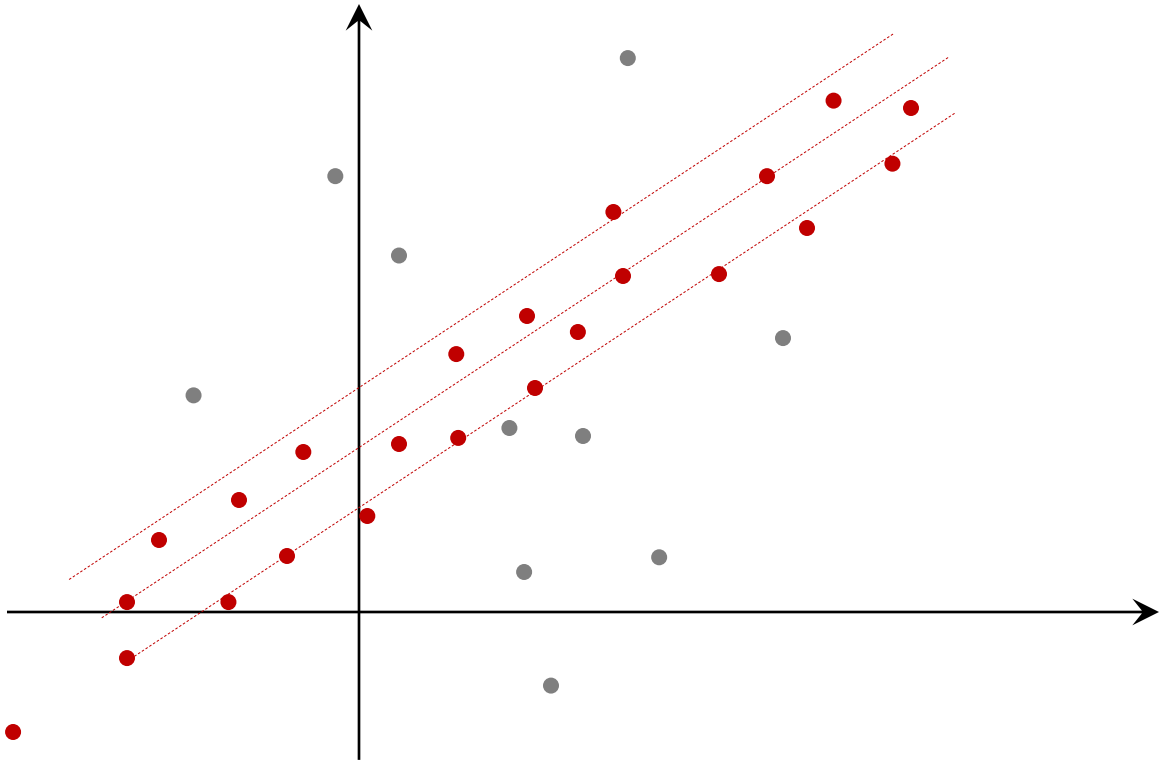


Required number of iterations with p success rate:



Required number of iterations with *p* success rate:

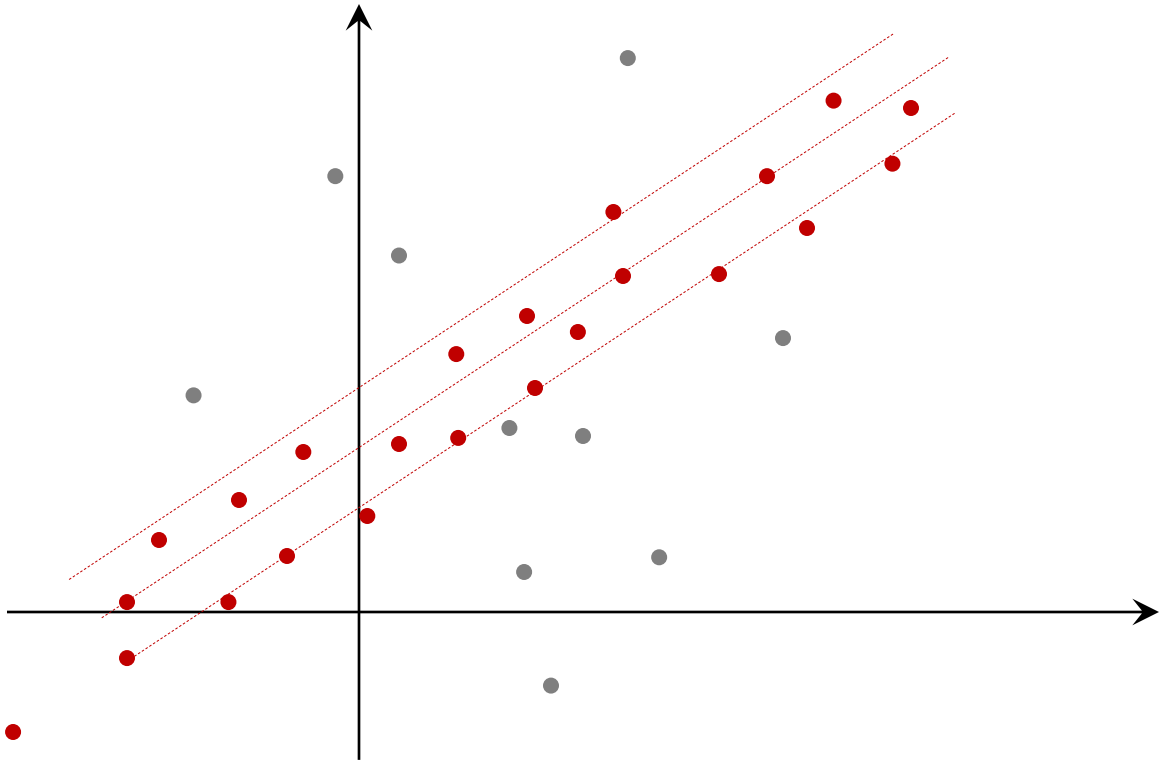
Probability of choosing an inlier $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$



Required number of iterations with *p* success rate:

Probability of choosing an inlier $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

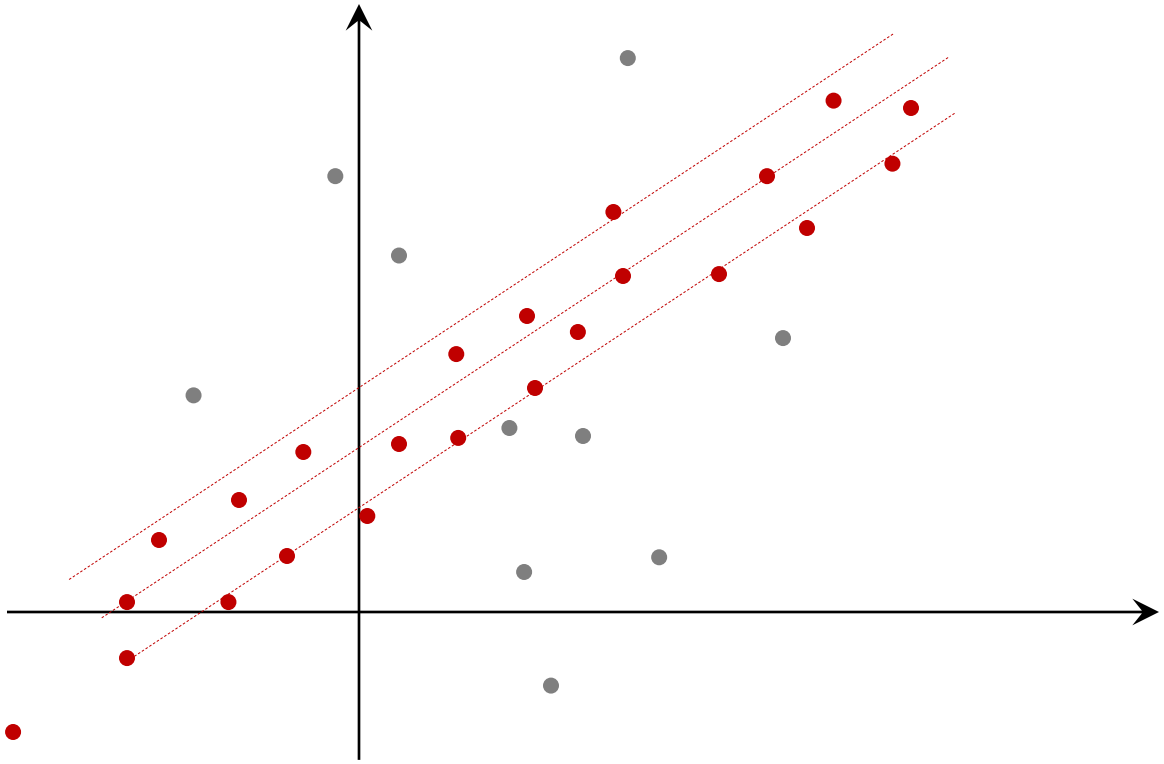


Required number of iterations with *p* success rate:

Probability of choosing an inlier $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

Probability of not building a correct model during *k* iterations: $\left(1 - w^n\right)^k$



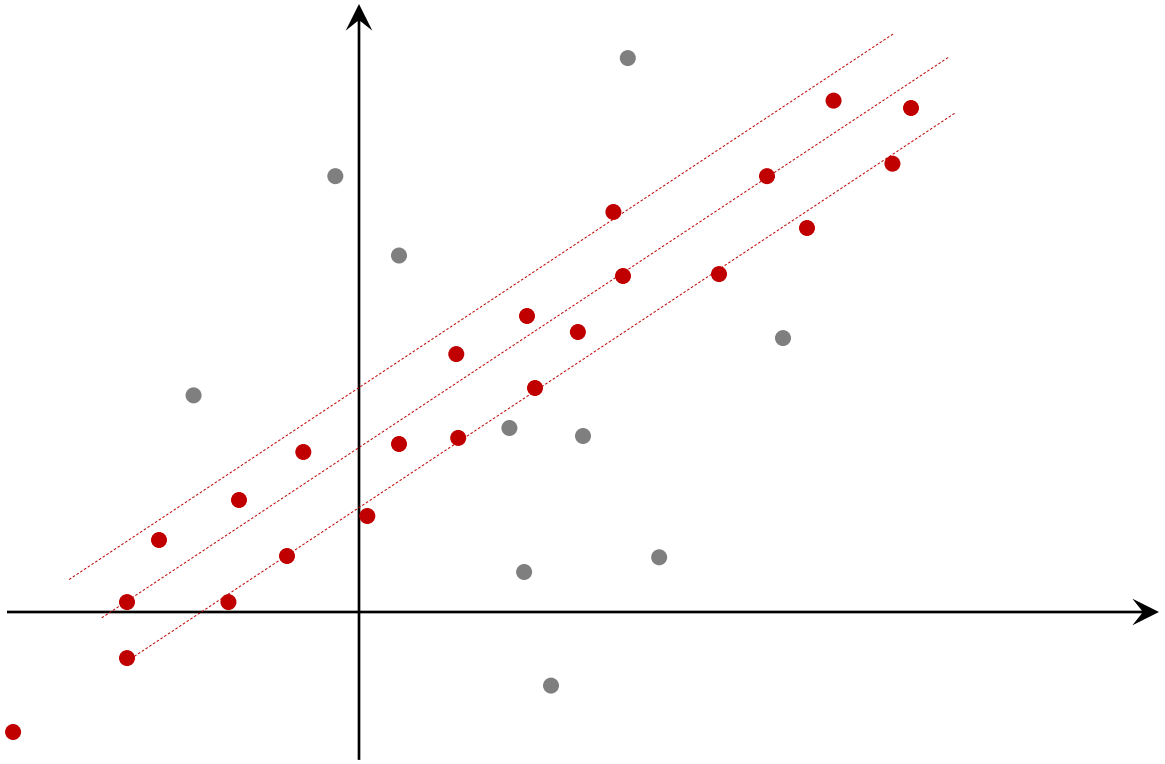
Required number of iterations with p success rate:

Probability of choosing an inlier $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model w^n where n is the number of samples to build a model.

Probability of not building a correct model during k iterations $\left(1-w^n\right)^k$

$\left(1-w^n\right)^k = 1-p$ where p is desired RANSAC success rate. $k = \frac{\log(1-p)}{\log\left(1-w^n\right)}$



Required number of iterations with p success rate:

$$k = \frac{\log(1-p)}{\log(1-w^n)} \qquad \text{where} w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$$

Probability of choosing an inlier $w = \frac{\#\text{ of inliers}}{\#\text{ of samples}}$

Probability of building a correct model: w^n where n is the number of samples to build a model.

Probability of not building a correct model during k iterations $\left(1-w^n\right)^k$

$\left(1-w^n\right)^k = 1-p$ where p is desired RANSAC success rate. $k = \frac{\log(1-p)}{\log(1-w^n)}$

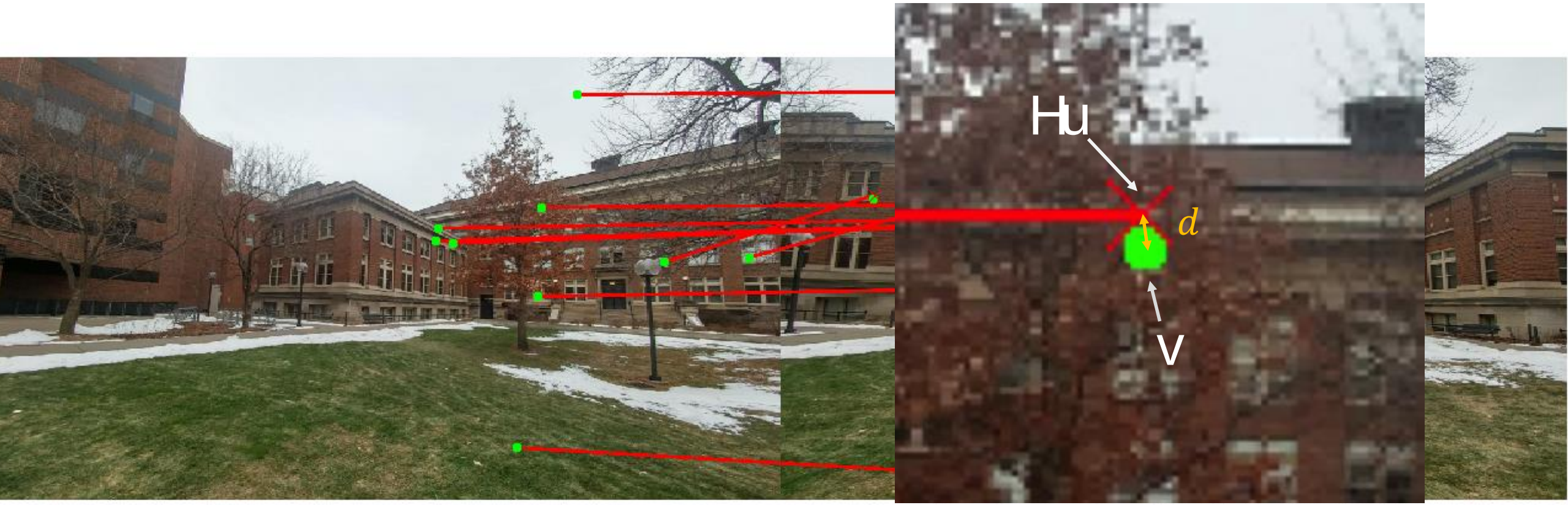
I_1 I_2

$$\left\{ \begin{array}{l} v_1 \leftrightarrow u_1 \\ v_2 \leftrightarrow u_2 \\ v_3 \leftrightarrow u_3 \\ v_4 \leftrightarrow u_4 \end{array} \right\} \rightarrow H$$

Homography computation

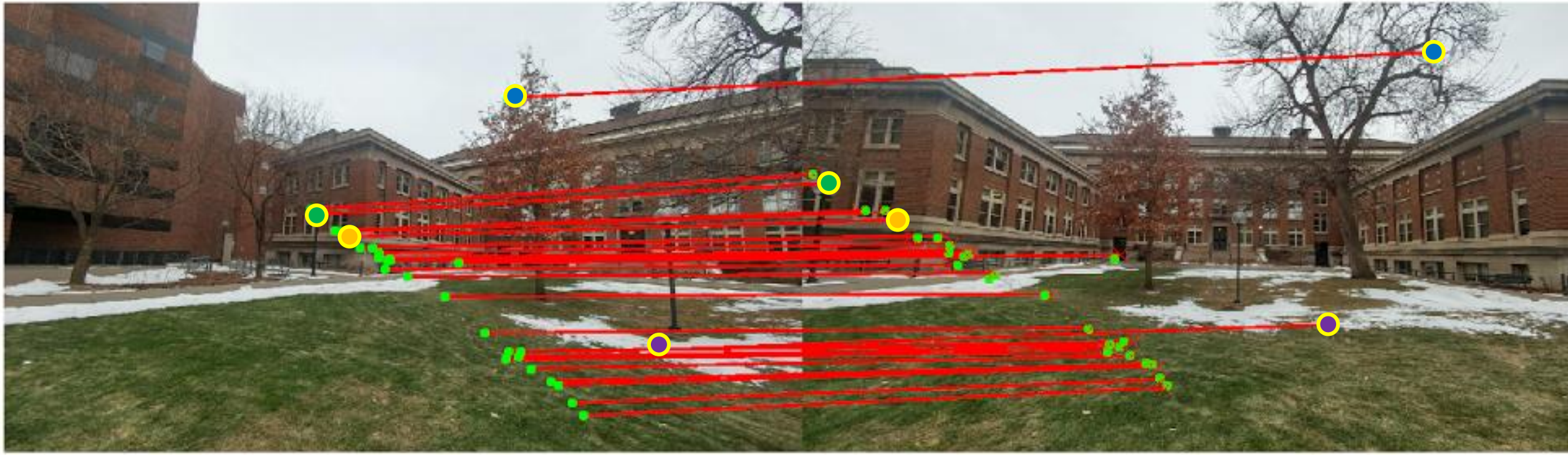
I_1 I_2

If the correspondence is bad, it has no prediction power!


Hu
d
V

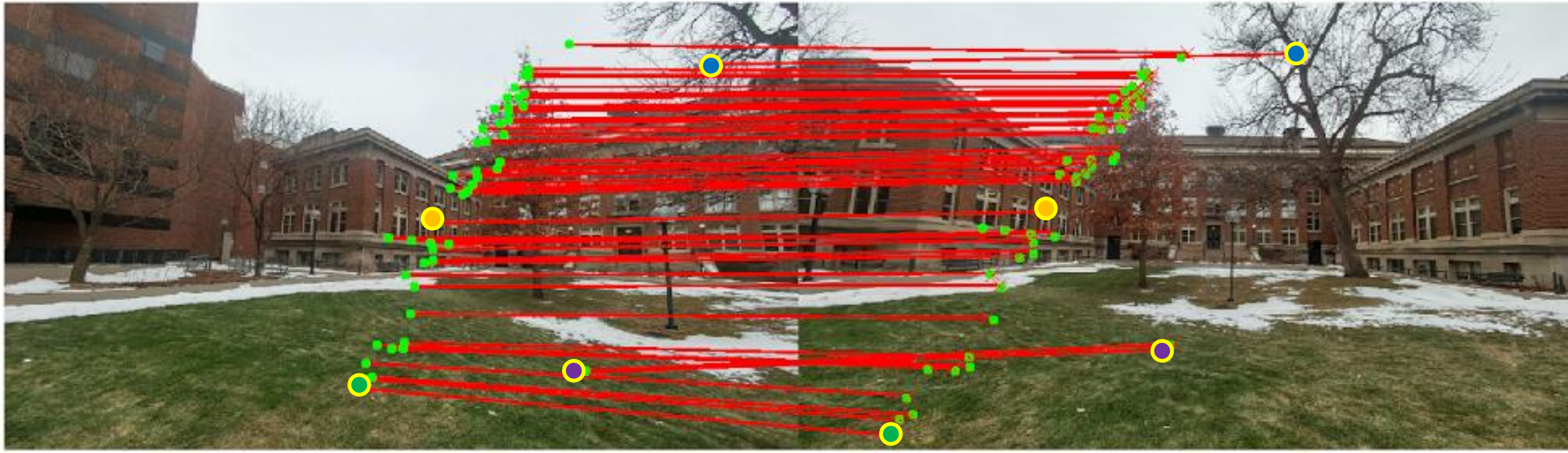
# of inliers: 16 out of 1865

# of inliers: 16 out of 1865



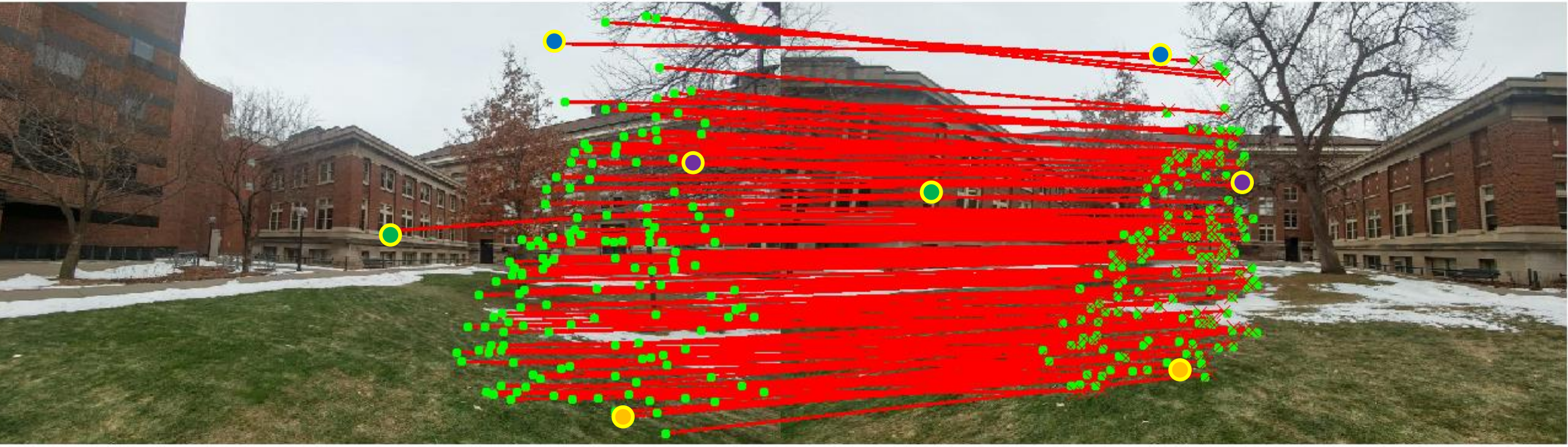
# of inliers: 36 out of 1865

# of inliers: 36 out of 1865



# of inliers: 57 out of 1865

# of inliers: 57 out of 1865



# of inliers: 216 out of 1865

# of inliers: 216 out of 1865

Euclidean Transform (Translation)


Homography