KENTECH
Korea Institute of Energy Technology

Advanced Computer Vision
Week 08

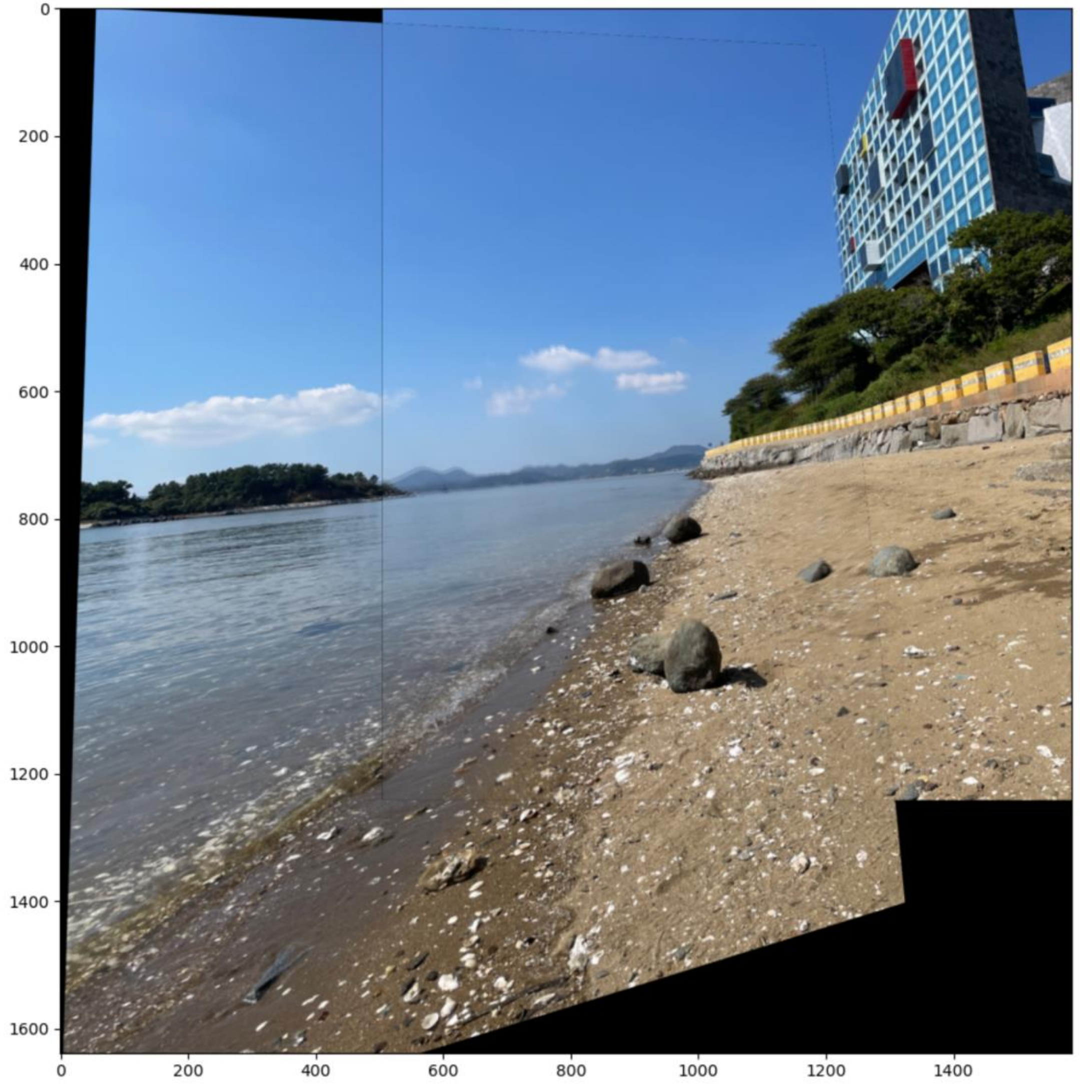
Oct. 28, 2022

Seokju Lee

Experiment 2: Panorama Stitching Using SIFT + RANSAC

→ Output results

Matching result after SIFT matching + RANSAC



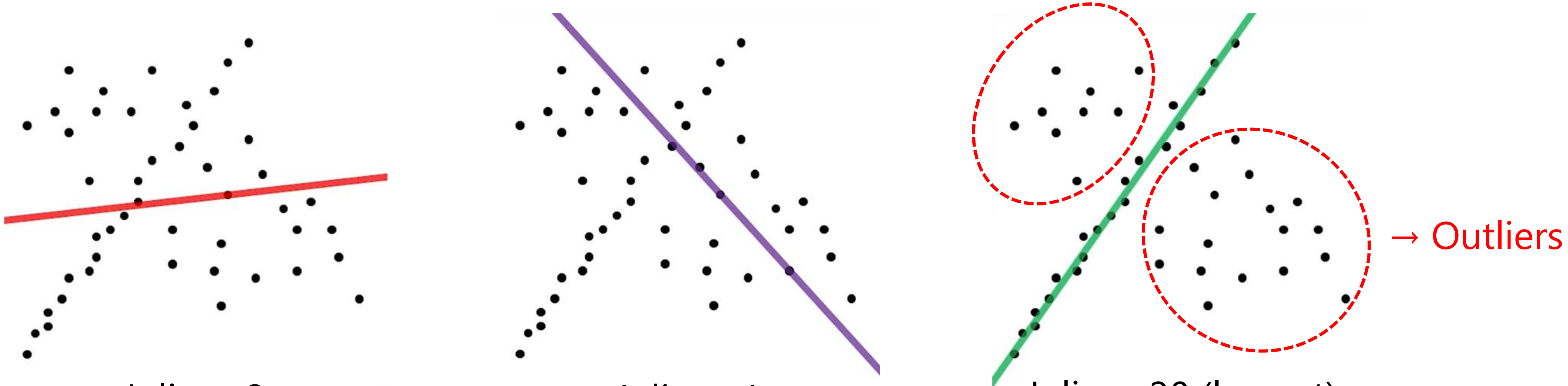
Warping result

RANdom SAmple Consensus

General RANSAC Algorithm :

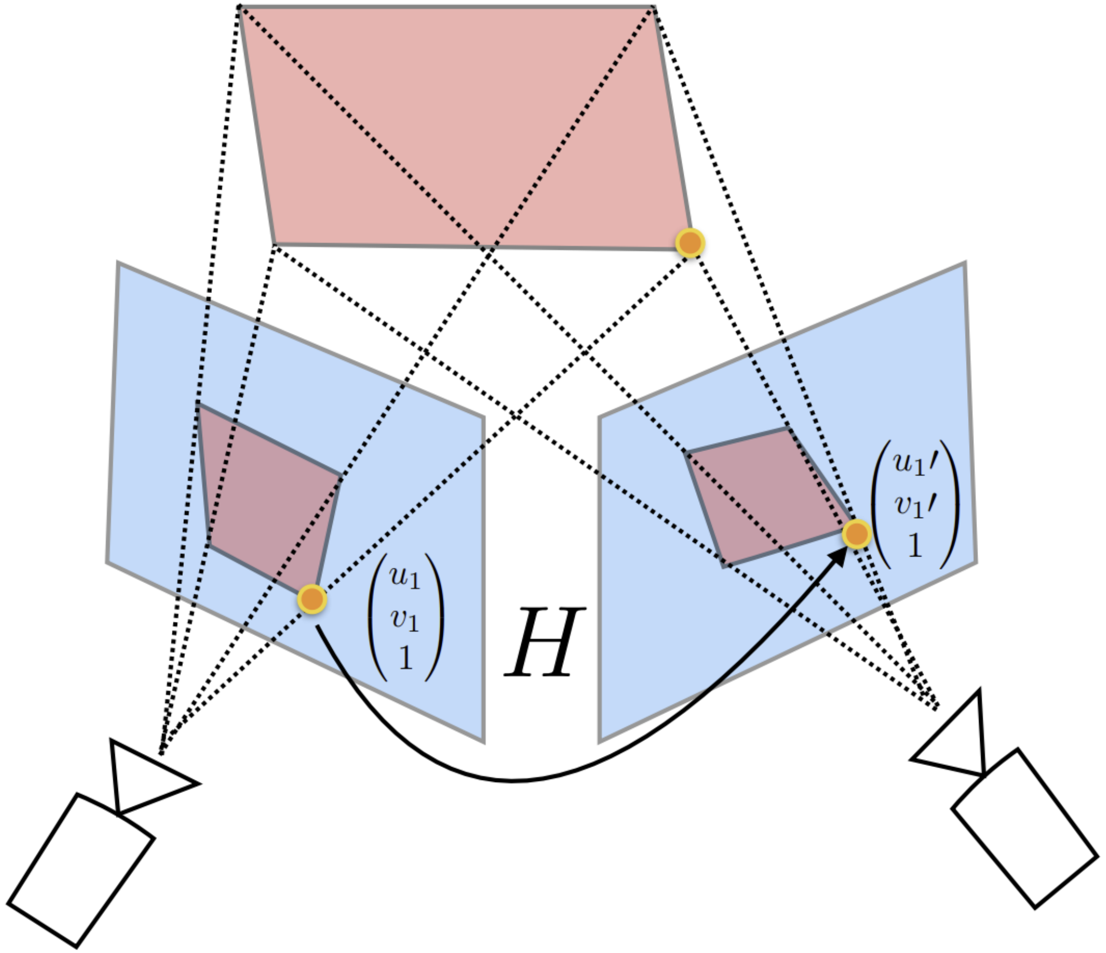
1. Randomly choose s samples. Typically s is the minimum samples to fit a model.
2. Fit the model to the randomly chosen samples.
3. Count the number M of data points (inliers) that fit the model within a measure of error ε.
4. Repeat Steps 1-3 N times.
5. Choose the model that has the largest number M of inliers.

ex) Line fitting :



Inliers: 2 Inliers: 4 Inliers: 20 (largest)

Finding Homography with Given 4 points Correspondence



$$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix}$$

1-to-1 mapping

$$H_{matrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}$$

[1] D. DeTone, et al., "Deep Image Homography Estimation", 2016.

Finding Homography with Given 4 points Correspondence

$$H_{projective} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \Rightarrow \begin{cases} x_i' = \dfrac{ax_i + by_i + c}{gx_i + hy_i + 1} \\ y_i' = \dfrac{dx_i + ey_i + f}{gx_i + hy_i + 1} \end{cases}$$

→ Solve for $\mathbf{p}$, if $A\mathbf{p} = 0$ → Least square method (**LSM**) or singular value decomposition (**SVD**)

$$\Rightarrow \begin{cases} x_i'(gx_i + hy_i + 1) = ax_i + by_i + c \\ y_i'(gx_i + hy_i + 1) = dx_i + ey_i + f \end{cases} \Rightarrow \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i'x_i & -y_i'y_i & -y_i' \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

→ In $\boldsymbol{H_{projective}}$, the **last** entry is defined only up to scales (**8-DoF**) → A 2D point has **2-DoF** to (x, y) components and each point-to-point **correspondence** accounts for **two constraints**. This is the reason why we need **4 points correspondences** for finding the homography.

[1] D. DeTone, et al., "Deep Image Homography Estimation", 2016.
[2] https://ai.stackexchange.com/questions/21042/how-do-you-find-the-homography-matrix-given-4-points-in-both-images

Python Codes

https://view.kentech.ac.kr/f088fa7f-874e-44bc-bd6d-6084b42dfdf7

- **Homography**

```
def homography(pairs):
    rows = []
    for i in range(pairs.shape[0]):
        p1 = np.append(pairs[i][0:2], 1)
        p2 = np.append(pairs[i][2:4], 1)
        row1 = [0, 0, 0, p1[0], p1[1], p1[2],
                -p2[1]*p1[0], -p2[1]*p1[1], -p2[1]*p1[2]]
        row2 = [p1[0], p1[1], p1[2], 0, 0, 0,
                -p2[0]*p1[0], -p2[0]*p1[1], -p2[0]*p1[2]]
        rows.append(row1)
        rows.append(row2)
    rows = np.array(rows)
    U, s, V = np.linalg.svd(rows)

    H = V[-1].reshape(3, 3)
    H = H/H[2, 2]
    return H
```

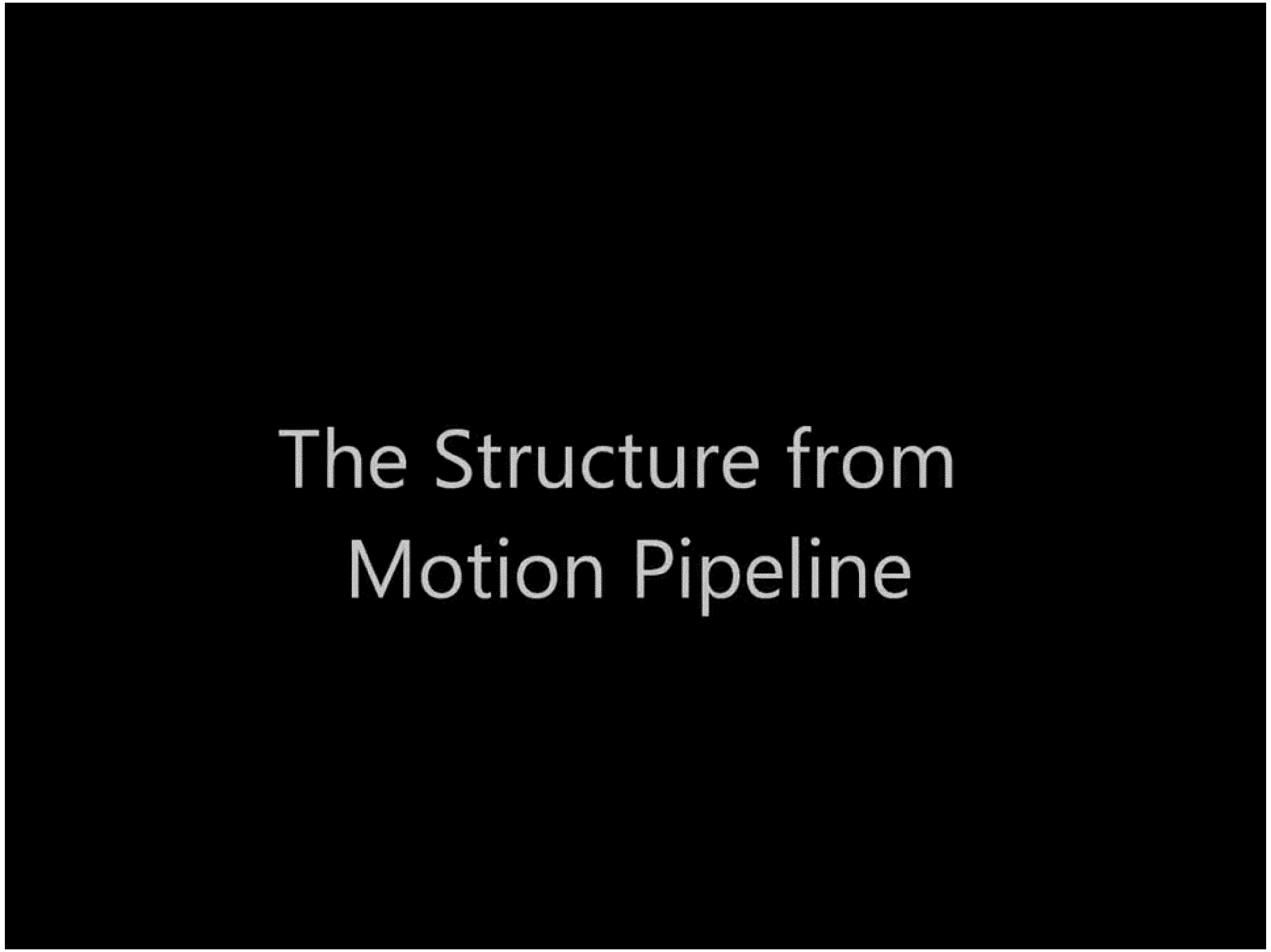
- **RANSAC**

```
def ransac(matches, threshold, iters):
    num_best_inliers = 0

    for i in range(iters):
        points = random_point(matches)
        H = homography(points)  # candidate

        if np.linalg.matrix_rank(H) < 3:
            continue  # Avoid dividing by zero.

        errors = get_error(matches, H)
        idx = np.where(errors < threshold)[0]
        inliers = matches[idx]

        num_inliers = len(inliers)
        if num_inliers > num_best_inliers:
            best_inliers = inliers.copy()
            num_best_inliers = num_inliers
            best_H = H.copy()

    return best_inliers, best_H
```

KENTECH
Korea Institute of Energy Technology

Structure-from-Motion (SfM)

3D Reconstruction Using SfM



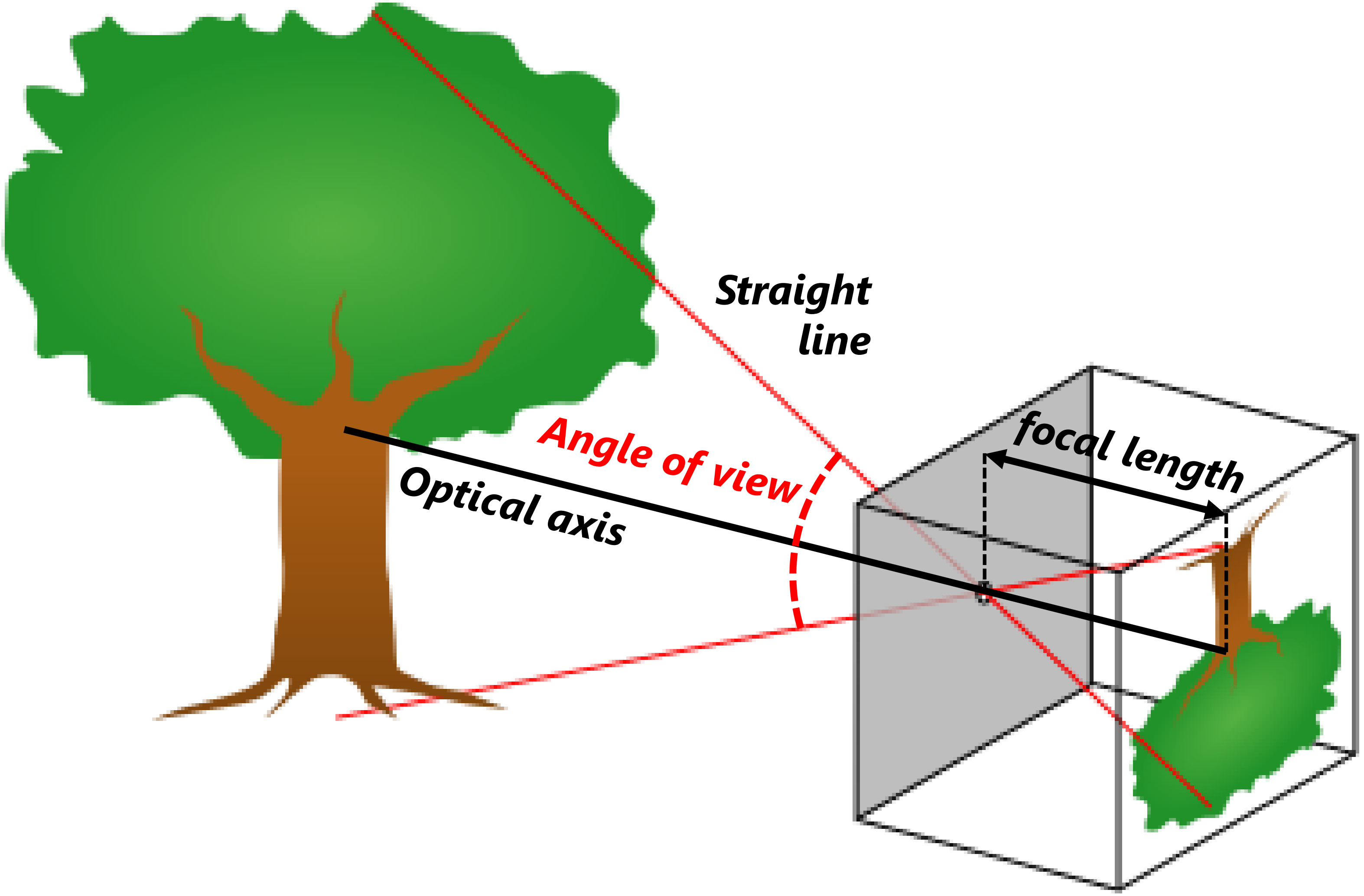
The Structure from Motion Pipeline

Contents

Multi-view 3D reconstruction

- Pinhole camera model
- Two-view geometry
- Structure-from-Motion (SfM)
- Epipolar geometry
- Essential matrix & Fundamental matrix
- Bundle adjustment
- …

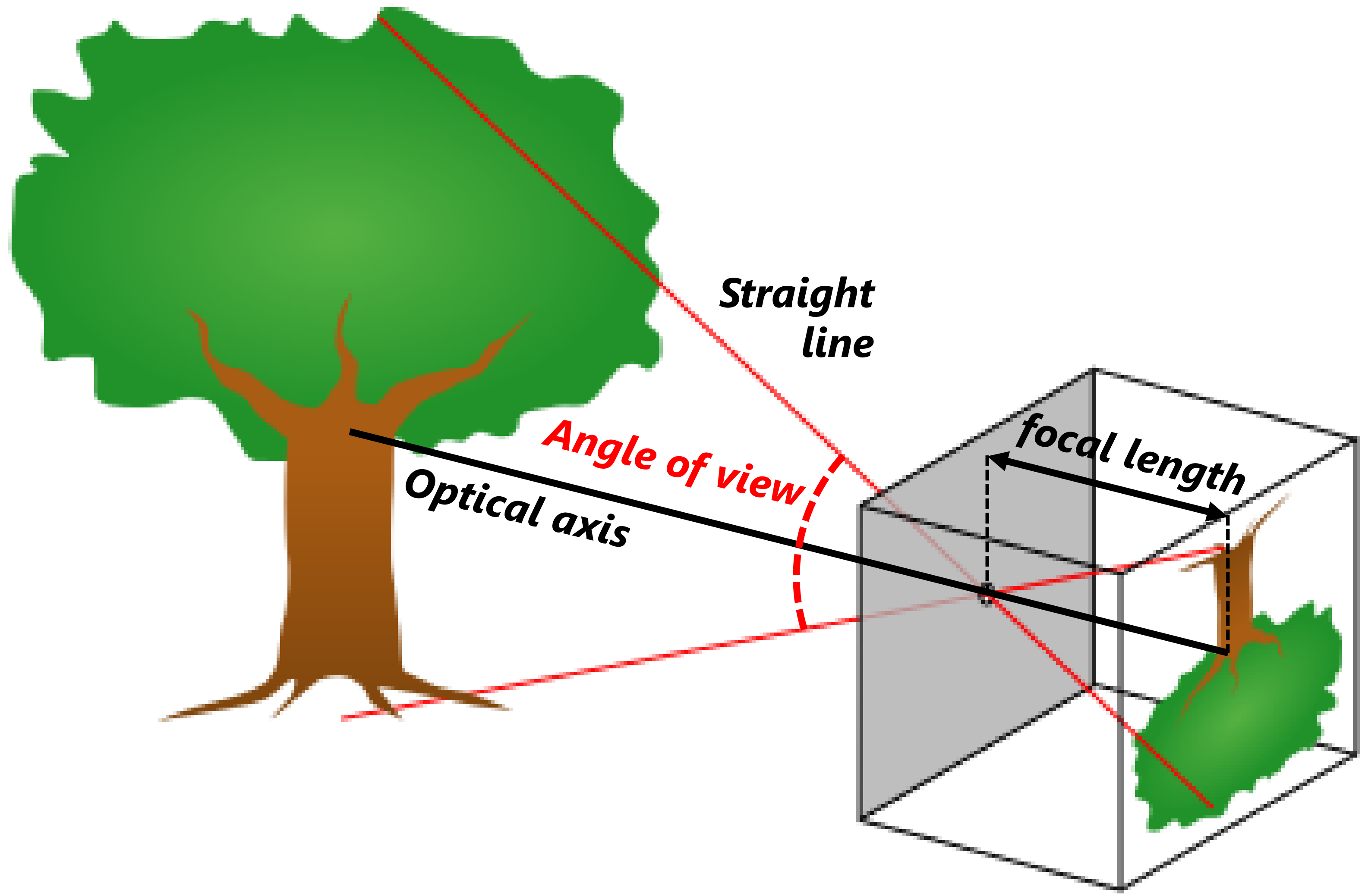
Pinhole Camera Model



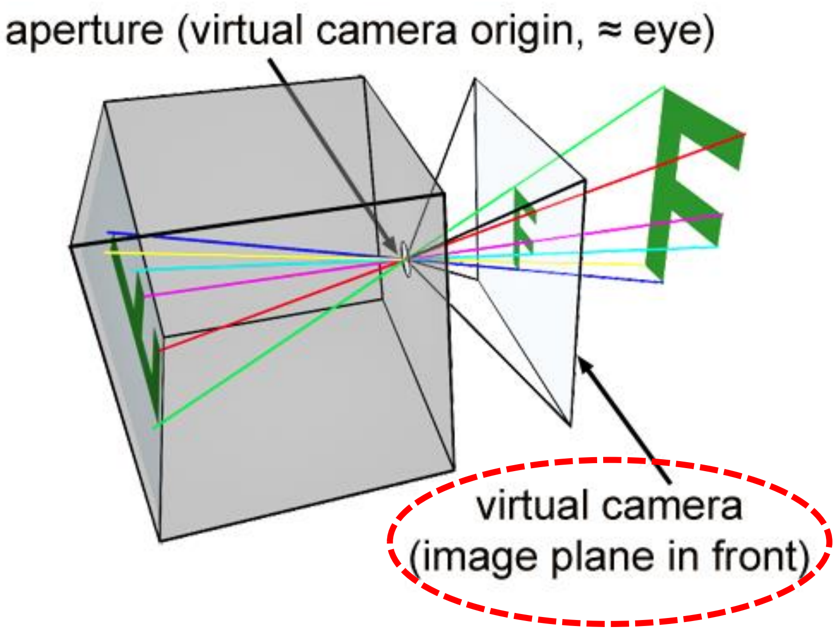
"Camera"

= Mapping from the **3D world** to the **2D image**

Pinhole Camera Model



+ Remind virtual image plane

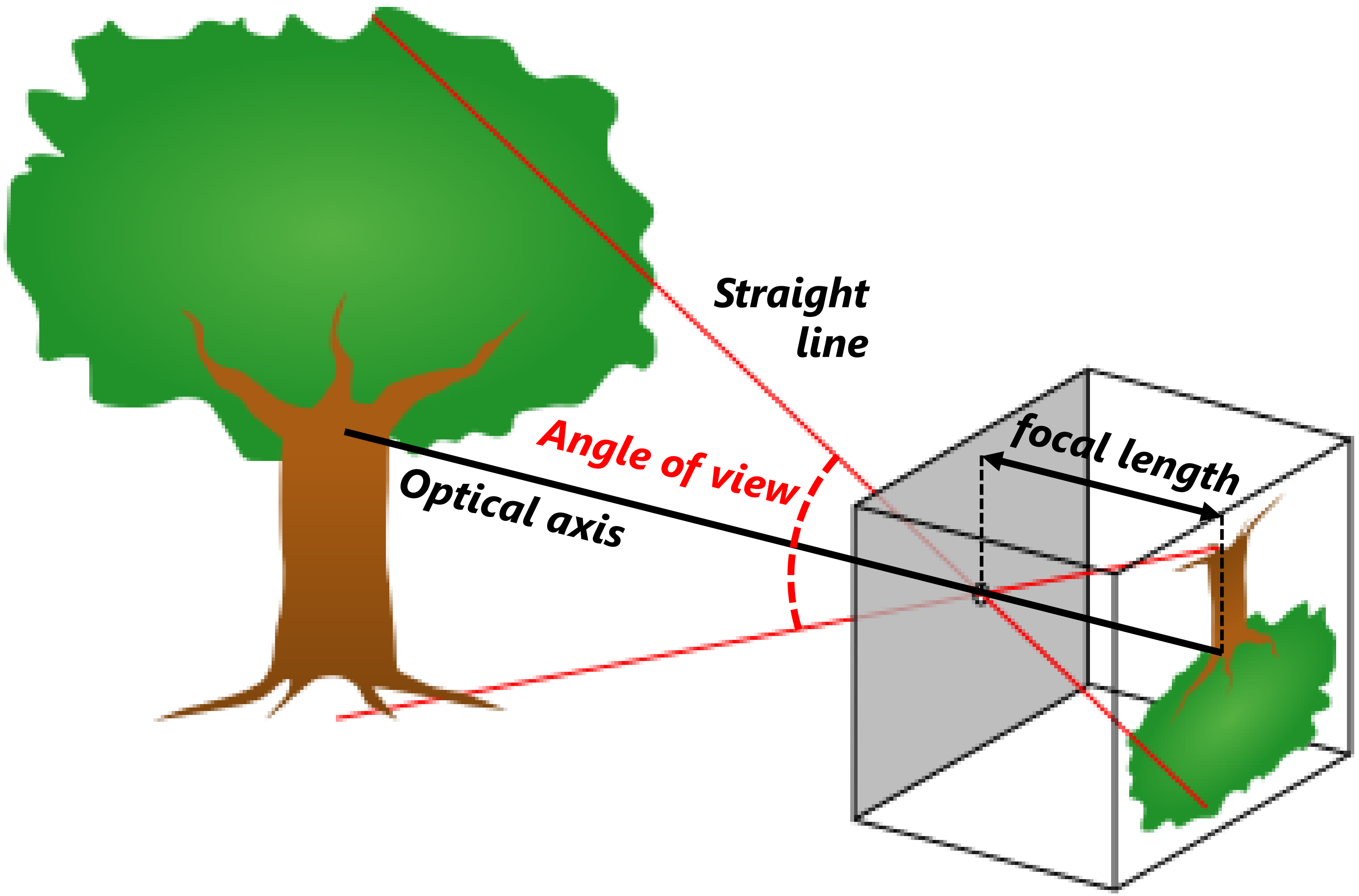


"Camera"

= Mapping from the **3D world**

to the **2D image**

Pinhole Camera Model



Homogeneous coordinates

$$x = \mathbf{P}\mathbf{X}$$

2D point **Camera matrix** **3D point**

"Camera"

= Mapping from the **3D world** to the **2D image**

Pinhole Camera Model

$$x = K[R|t]\mathbf{X} = \mathbf{PX}$$

→ For each corresponding point i in the image:

$$\underbrace{\begin{bmatrix} u^{(i)} \\ v^{(i)} \\ 1 \end{bmatrix}}_{\substack{\text{Homogeneous} \\ \text{image coordinates} \\ 3 \times 1}} \equiv \underbrace{\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}}_{\substack{\text{Camera} \\ \text{projection matrix} \\ 3 \times 4}} \underbrace{\begin{bmatrix} x_w^{(i)} \\ y_w^{(i)} \\ z_w^{(i)} \\ 1 \end{bmatrix}}_{\substack{\text{Homogeneous} \\ \text{world coordinates} \\ 4 \times 1}}$$

Pinhole Camera Model

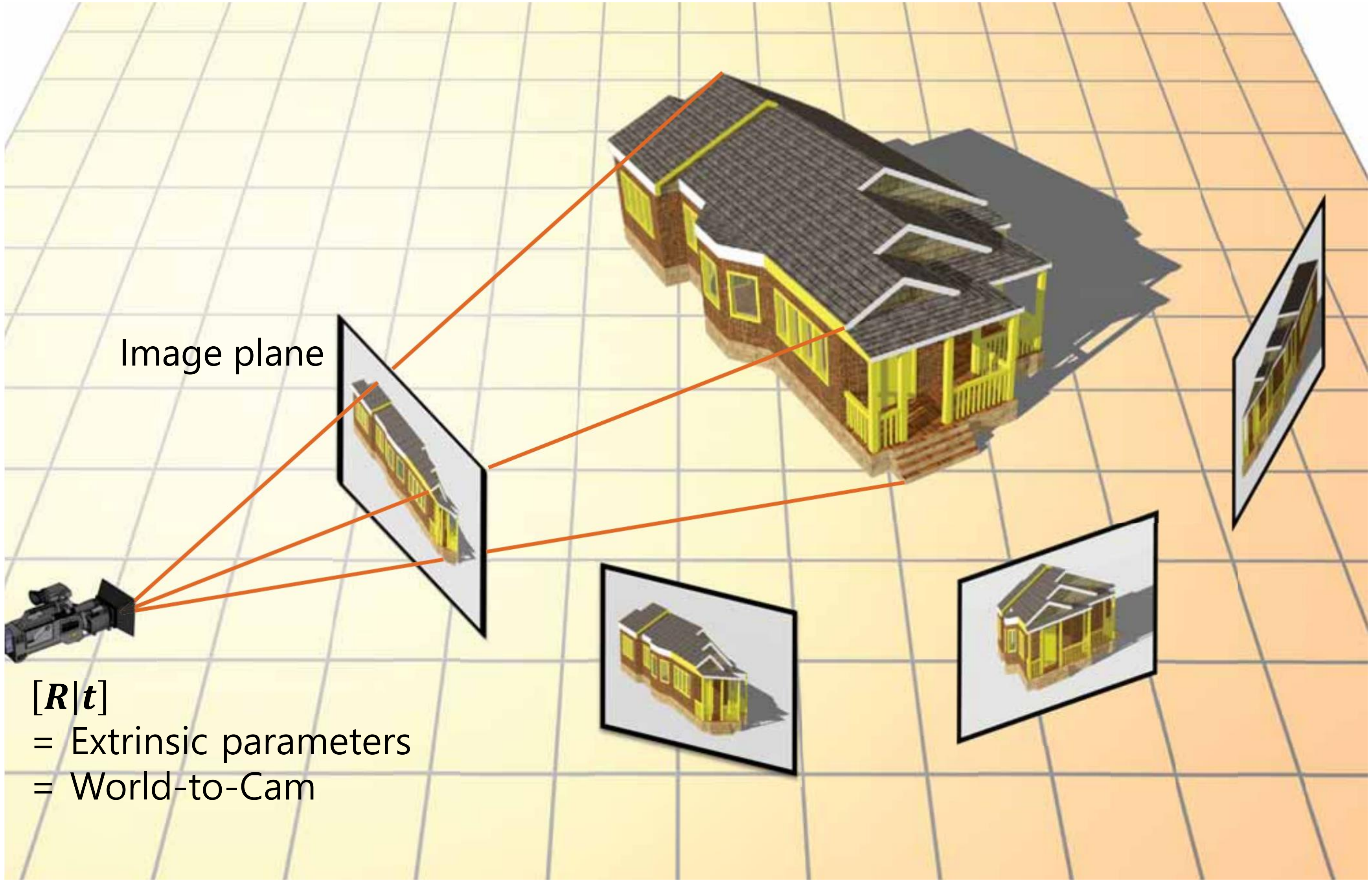
$$x = K[R|t]\mathbf{X} = \mathbf{PX}$$

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \text{skew\_c}f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= K[R|t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

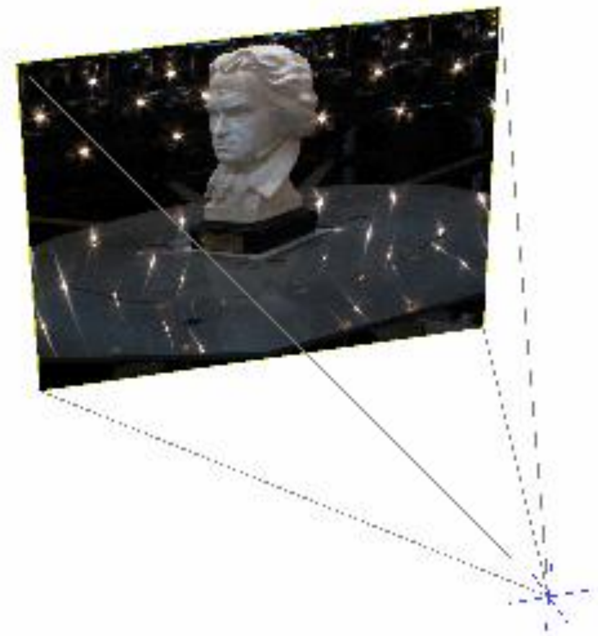
- $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$: 3D point in the world coordinate
- $[\boldsymbol{R}|\boldsymbol{t}]$: extrinsic parameters to convert the world coordinate into the camera coordinate
- $\boldsymbol{K}$: intrinsic parameters to represent the camera characteristics
- $\boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$: camera projection matrix
- $(\boldsymbol{x}, \boldsymbol{y})$: 2D pixel location in the image plane
- $\boldsymbol{s}$: scale factor

Multi-View 3D Reconstruction



http://www.cs.cmu.edu/~yaser/TutorialSlides-Integrated-small.pdf

Multi-View 3D Reconstruction



https://vision.in.tum.de/research/image-based_3d_reconstruction/multiviewreconstruction

Structure-from-Motion Pipeline

Given only the 2D multi-view images of a scene,

recover the underlying 3D **structure** and the camera **motion**.

Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

Structure-from-Motion Pipeline

Q1. How to find **2D-3D points** to reconstruct?

Q2. How to find an **optimal** 3D structure and camera poses for multiple view?

Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

Structure-from-Motion Pipeline

Q1. How to find **2D-3D points** to reconstruct?

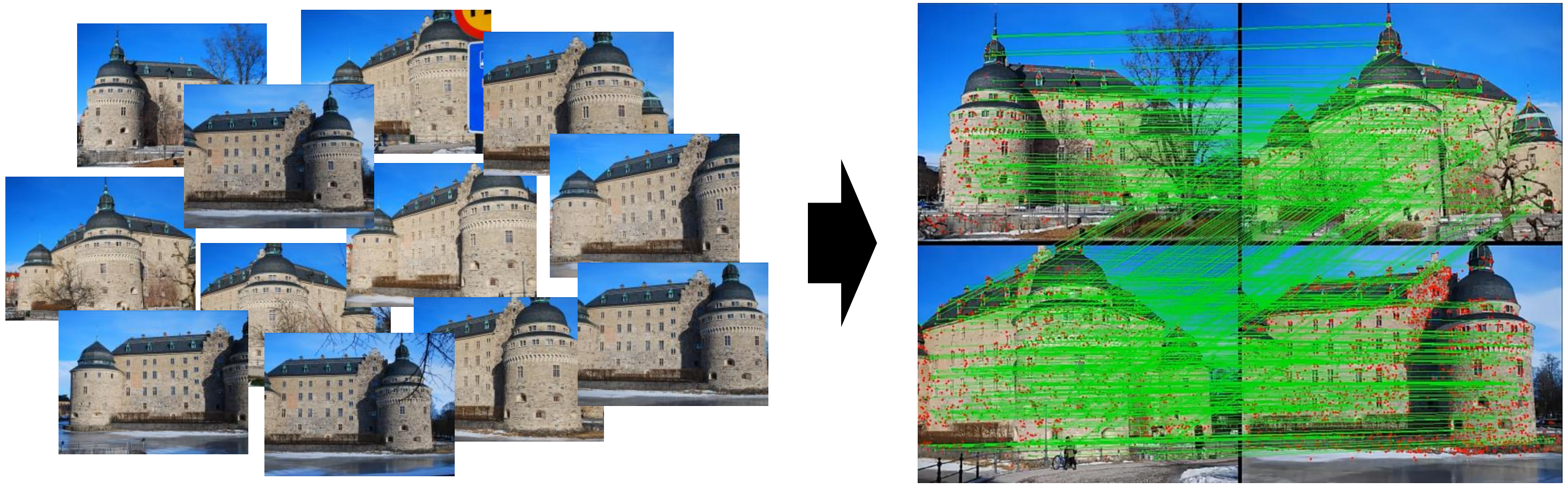
Q2. How to find an **optimal** 3D structure and camera poses for multiple view?

Schonberger, et al., “Structure-from-Motion Revisited”, CVPR 2016.

Structure-from-Motion Pipeline

Correspondence search

- Feature extraction
- Matching
- Geometric verification



A set of images　　　*Feature extraction & matching (graph)*

Schonberger, et al., “Structure-from-Motion Revisited”, CVPR 2016.

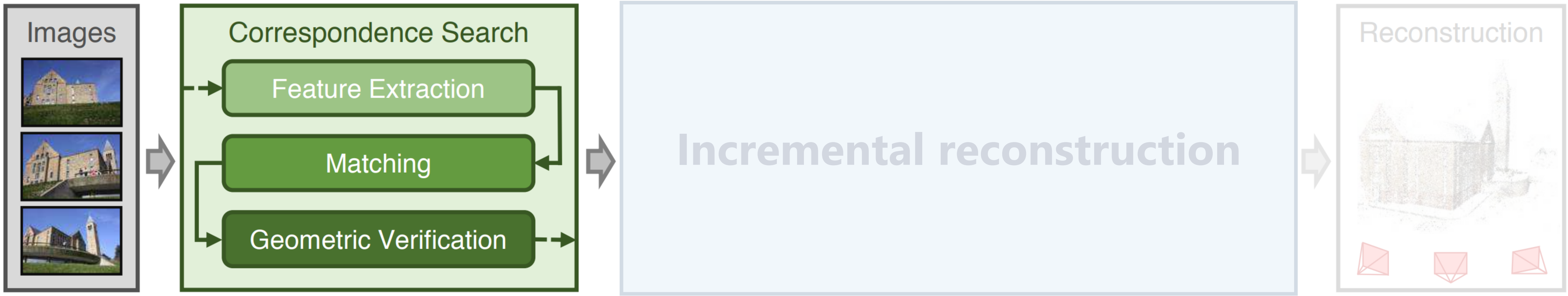
Structure-from-Motion Pipeline

c.f.) Homography matrix

Correpondence search

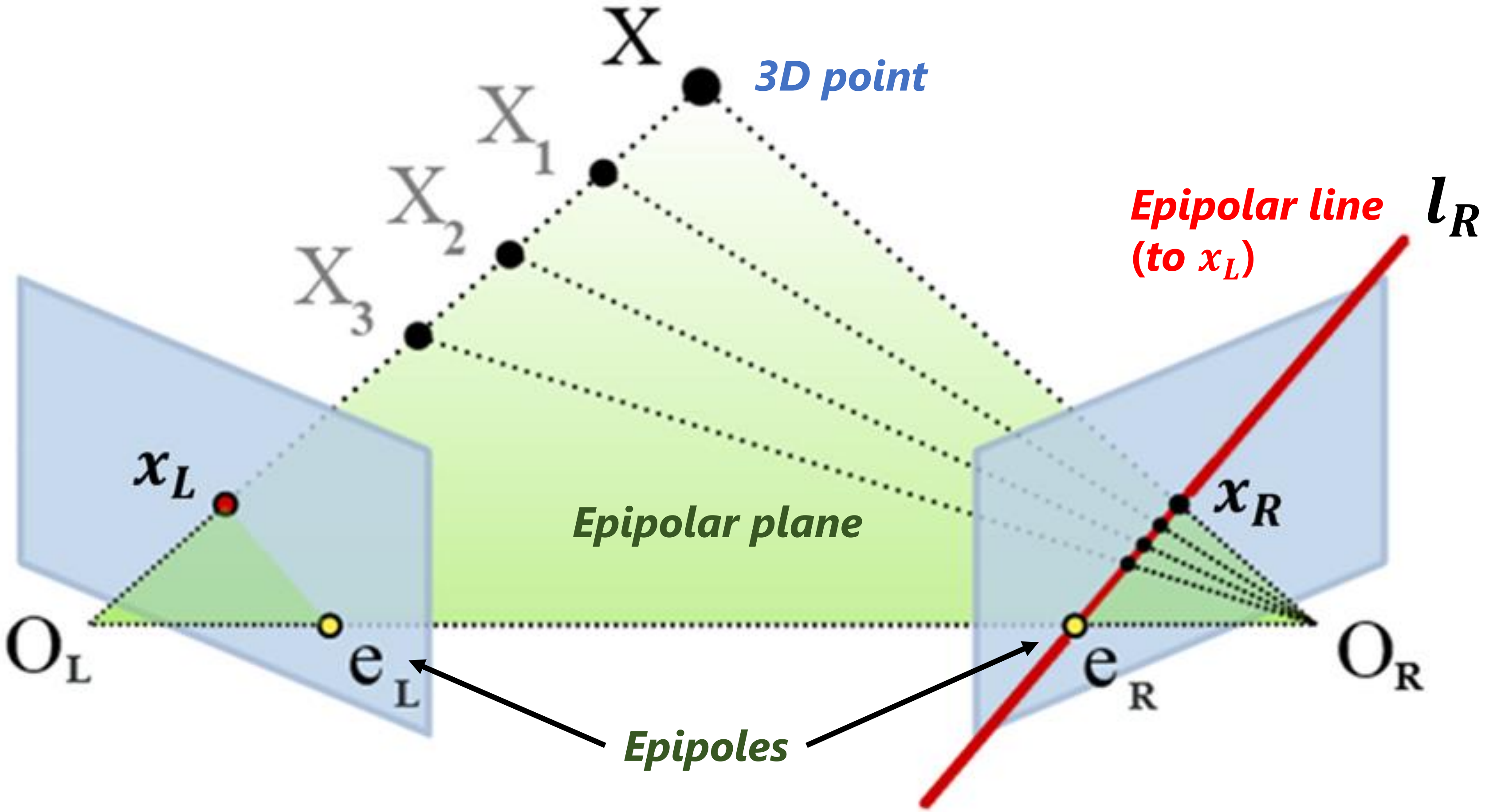
- Feature extraction
- Matching
- **Geometric verification**

- Epipolar geometry describes the relation of moving cameras **(5 or 8-point algorithm)** through the essential matrix, $\mathbf{E} \in \mathbb{R}^{3\times3}$, or the fundamental matrix, $\mathbf{F} \in \mathbb{R}^{3\times3}$
- If estimated $\mathbf{E}$ projects a sufficient number of features between the images, it is verified! **(RANSAC)**



Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

Epipolar Geometry

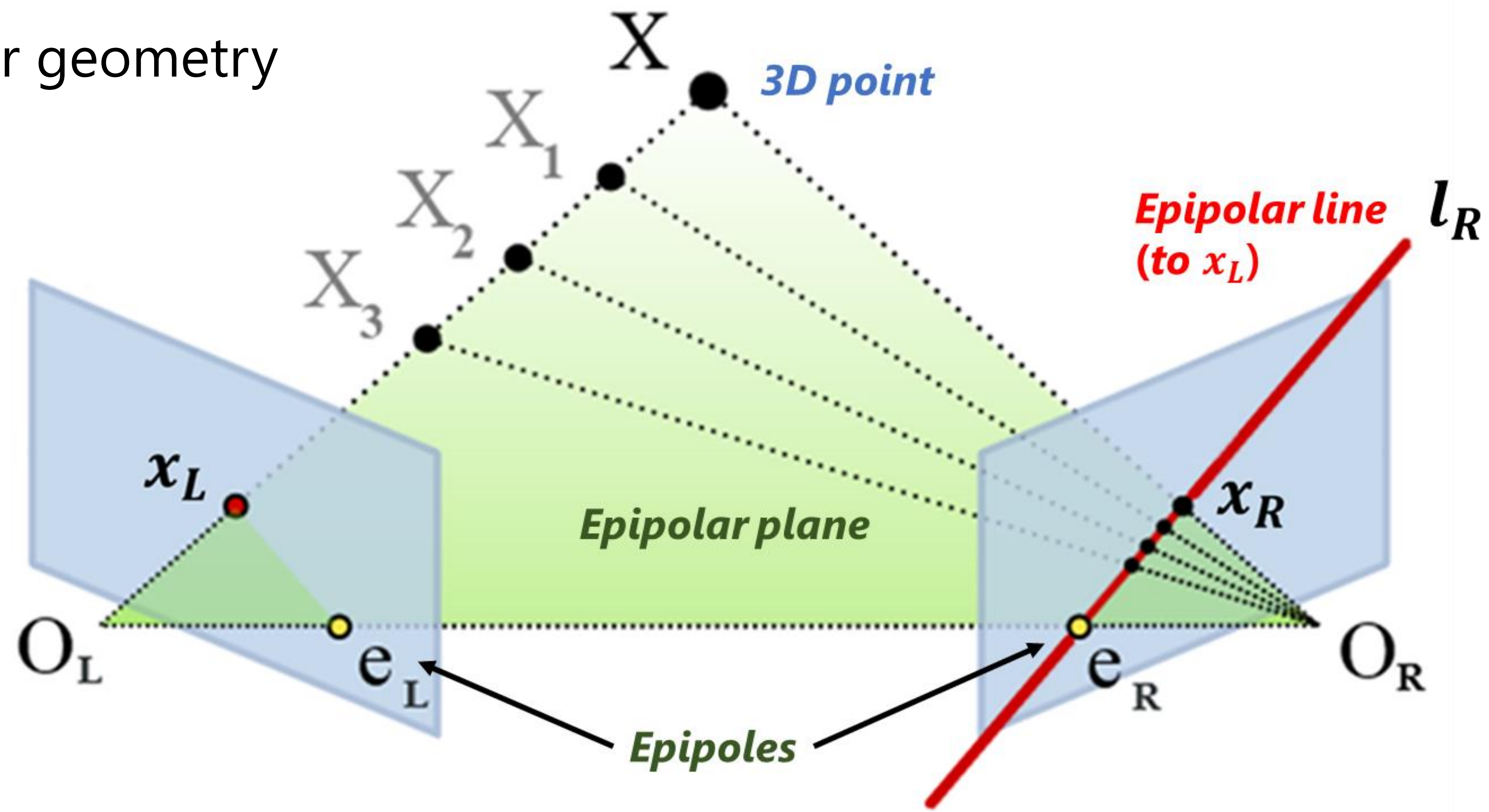


Potential matches for $\boldsymbol{x_L}$ are on the epipolar line $\boldsymbol{l_R}$

Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

Essential Matrix

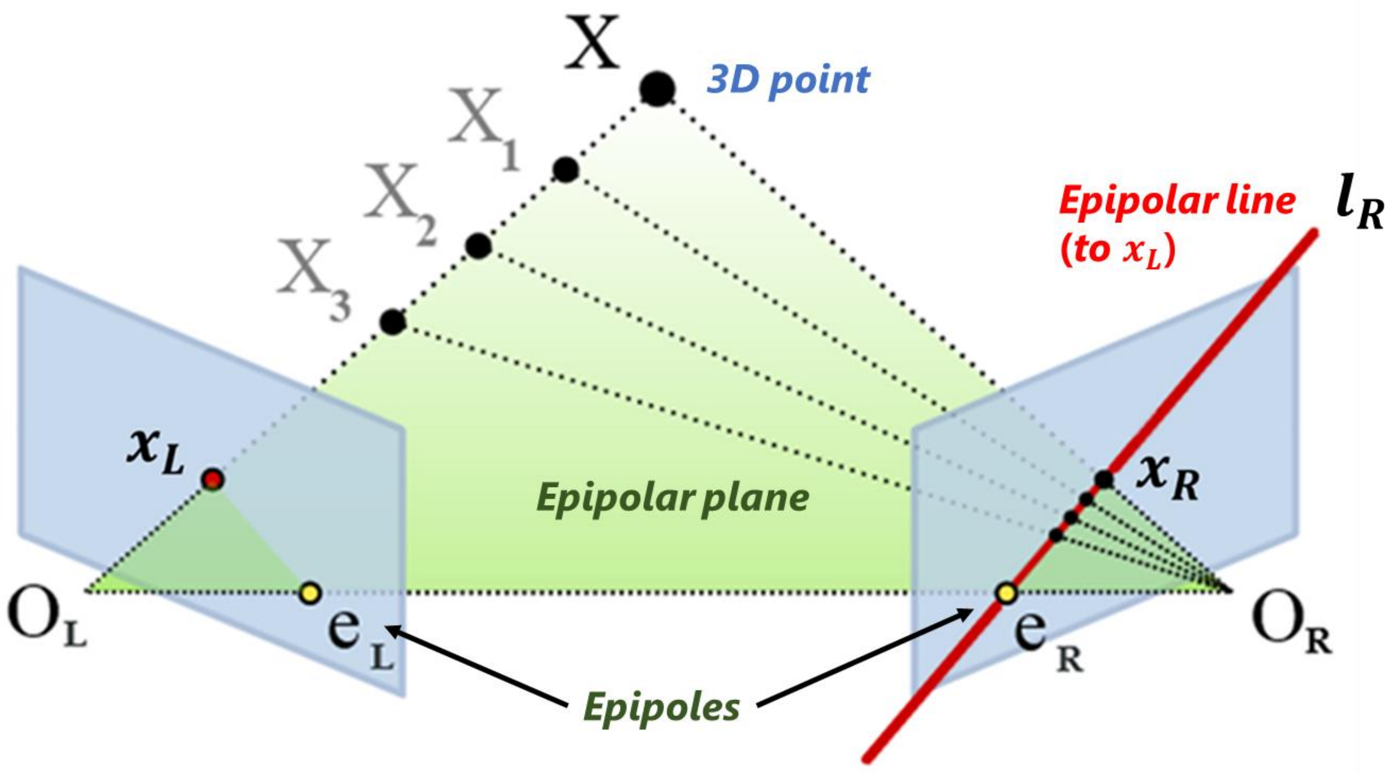
: Encodes epipolar geometry



Given a point $\boldsymbol{x_L}$ in one image, multiplying by the **essential matrix** $\mathbf{E} \in \mathbb{R}^{3\times3}$ will tell us the epipolar line in the right view: $\mathbf{E}\boldsymbol{x_L} = \boldsymbol{l_R}$

Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

Epipolar Constraint



$ax + by + c = 0$ in vector form $l = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$.

If the point $\boldsymbol{x_R}$ is on the epipolar line $\boldsymbol{l_R}$,

$$\boldsymbol{x_R}^\top \boldsymbol{l_R} = 0$$

Since $\mathbf{E}\boldsymbol{x_L} = \boldsymbol{l_R}$,

$$\boldsymbol{x_R}^\top \mathbf{E}\boldsymbol{x_L} = 0$$

Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

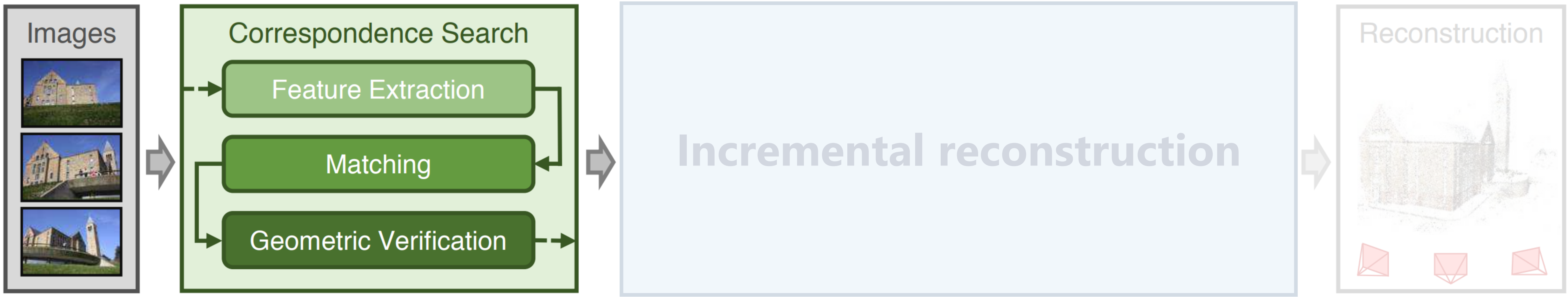
Structure-from-Motion Pipeline

Geometric verification

- **5-point** / **8-point** algorithm:

  Estimating the **essential matrix** from the feature point correspondences.

  → Direct linear transform: $\boldsymbol{x_R}^{\top}\mathbf{E}\boldsymbol{x_L} = 0 \rightarrow A\mathbf{p} = 0 \rightarrow$ Use SVD to solve!



Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

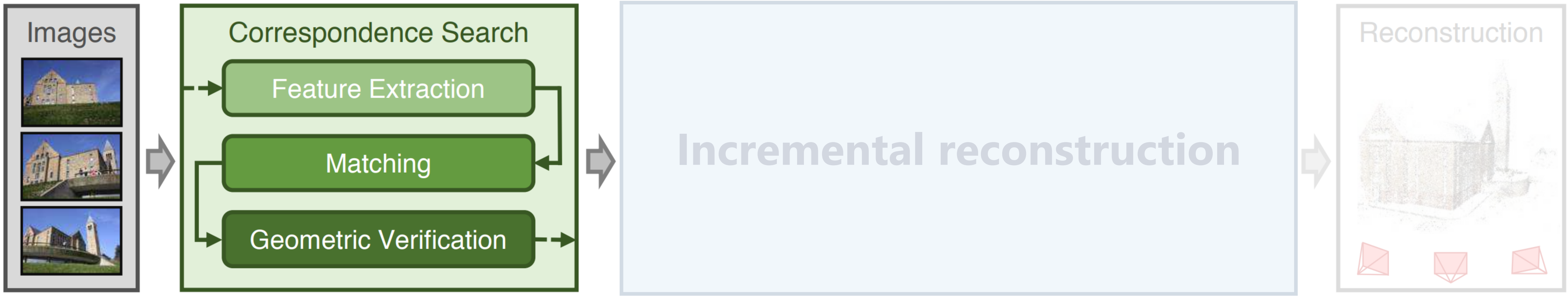
Structure-from-Motion Pipeline

Geometric verification

- **5-point** / **8-point** algorithm: $\boxed{\boldsymbol{x_R}^\top \mathbf{E}\boldsymbol{x_L} = 0 \rightarrow A\mathbf{p} = 0}$

  Sample at least **5** or **8** points and compute the **essential matrix**.

  → **RANSAC** to discriminate **inliers/outliers** and the **best** essential matrix!

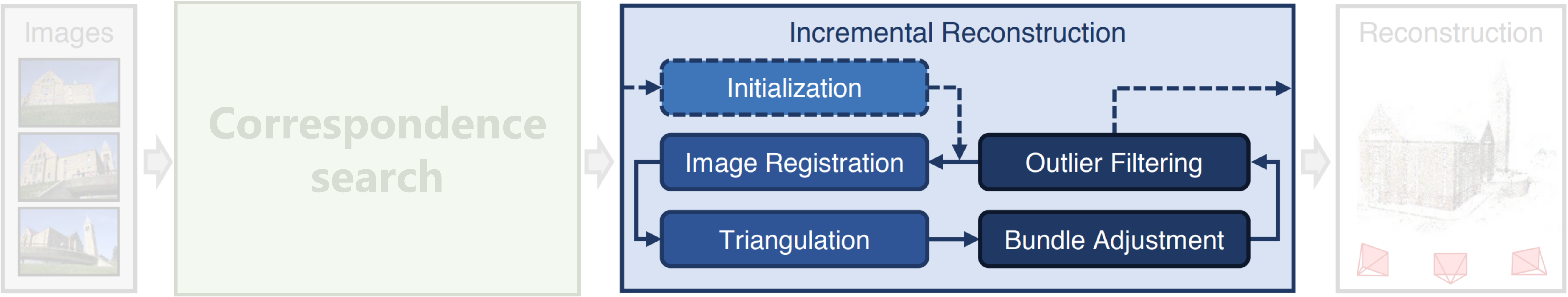


Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.

Next Contents

Incremental reconstruction

- Starting from **two** views, aggregate **more** views to **refine** the estimation.
- Camera initialization
- Triangulation + Bundle adjustment



Schonberger, et al., "Structure-from-Motion Revisited", CVPR 2016.