

# 3D Vision 4

Week 10

Multiple-view Geometry: Structure-from-Motion

Multiple-view Geometry: Novel View Synthesis

# Announcements

- **Assignment 3** due 11:59pm Friday 17 May
  - **Zero** marks if either report or code submitted late (unless extension)
    - Submit early; you can always resubmit an updated version later
    - Depending on your internet connection and load on the TurnItIn servers, uploading can sometimes be slow, so please factor this into your submission schedule
  - Submit your report (PDF) and code (ZIP file) **separately under the correct tab** in the submission box
  - Follow the instructions under Submission Requirements

# Weekly Study Plan: Overview

Wk	Starting	Lecture	Lab	Assessment
1	19 Feb	Introduction	X	
2	26 Feb	Low-level Vision 1	1	
3	4 Mar	Low-level Vision 2	1	
		Mid-level Vision 1		
4	11 Mar	Mid-level Vision 2	1	CLab1 report due Friday
		High-level Vision 1		
5	18 Mar	High-level Vision 2	2	
6	25 Mar	High-level Vision 3 <sup>1</sup>	2	
	1 Apr	Teaching break	X	
	8 Apr	Teaching break	X	
7	15 Apr	3D Vision 1	2	CLab2 report due Friday
8	22 Apr	3D Vision 2	3	
9	29 Apr	3D Vision 3	3	
10	6 May	3D Vision 4	3	
		Mid-level Vision 3		
11	13 May	High-level Vision 4	X	CLab3 report due Friday
12	20 May	Course Review	X	

# Weekly Study Plan: Part B

Wk	Starting	Lecture	By
7	15 Apr	3D vision: introduction, camera model, single-view geometry	Dylan
8	22 Apr	3D vision: camera calibration, two-view geometry (homography)	Dylan
9	29 Apr	3D vision: two-view geometry (epipolar geometry, triangulation, stereo)	Dylan
10	6 May	3D vision: multiple-view geometry	Weijian
		Mid-level vision: optical flow, shape-from-X	Dylan
11	13 May	High-level vision: self-supervised learning, detection, segmentation	Dylan
12	20 May	Course review	Dylan

# Outline

1. Multiple-view Geometry: Structure-from-Motion
2. Multiple-view Geometry: Novel View Synthesis

# Multi-View Geometry



Structure From Motion (SfM)

Extending two-view epipolar geometry to multiple views



Neural Radiance Fields (NeRFs),  
ECCV 2020

Solves correspondences + triangulation implicitly



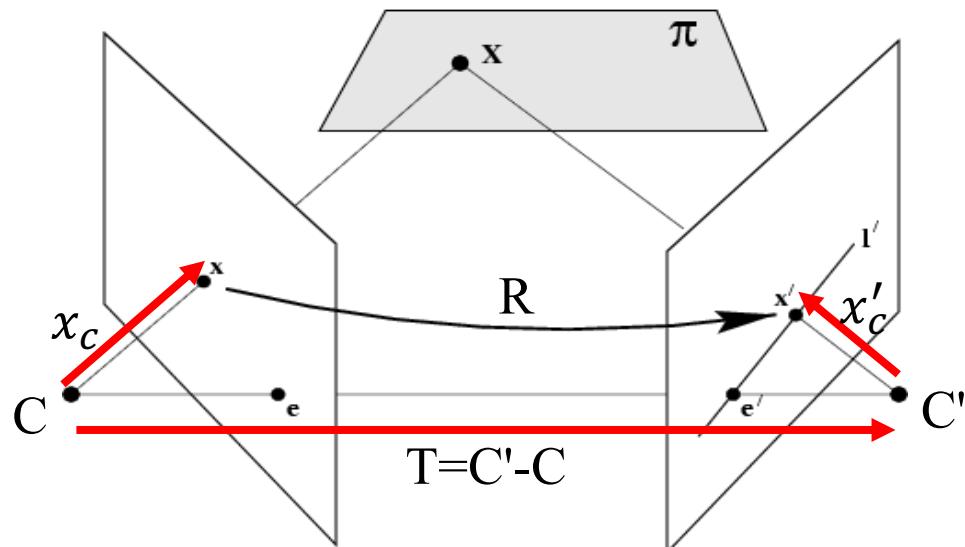
DUSt3R: Geometric 3D Vision Made Easy, CVPR 2024

Deep-learning based method, unify all 3D vision tasks

# Structure From Motion

- Epipolar Geometry

Geometric relationship between **two views** of the same scene captured from different viewpoints.



## Epipolar Geometry

- i) Fundamental Matrix

$$x'^T_{img} F x_{img} = 0$$

$$F = K'^{-T} E K^{-1} \quad K: \text{intrinsics mat.}$$

- ii) Essential Matrix

$$x'^T_c E x_c = 0 \quad R: \text{rotation mat.}$$

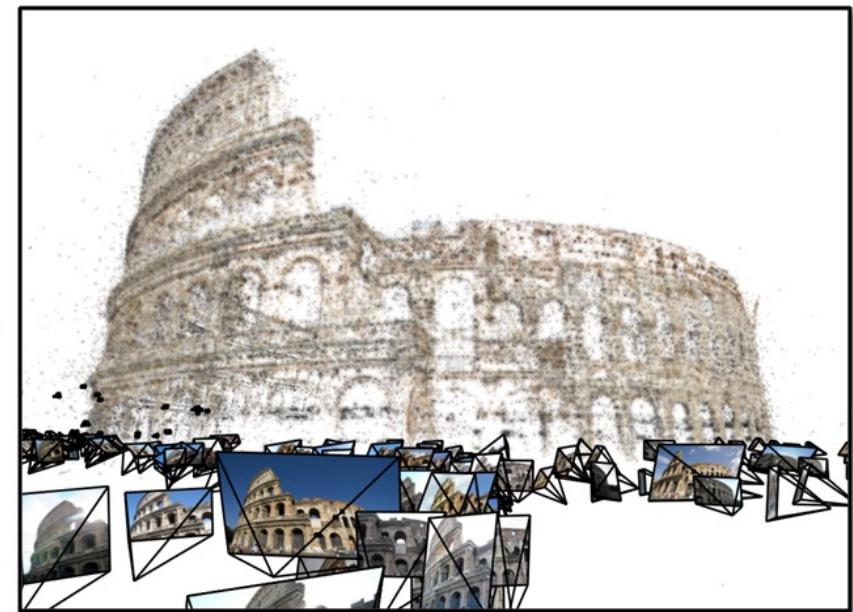
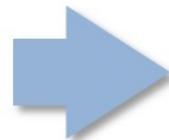
$$E = [t]_x R \quad t: \text{translation mat.}$$

- iii) Triangulation: Initial 3D points

# Structure From Motion

- What is structure from motion?

Structure from Motion is a 3D vision technique that **builds 3D structures** from multiple 2D images.



# Structure From Motion

- What is structure from motion?

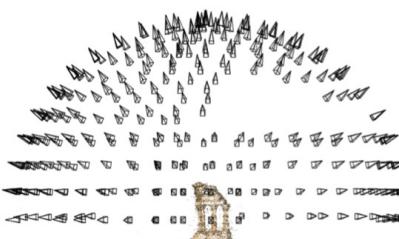
**Input:** multiple camera views (or video)

**Output:**

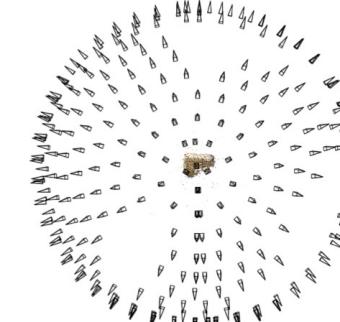
- 1) **3D structure**: captured scene (3D location for 2D pixel)
- 2) **camera "motion"**: camera parameters (intrinsic and extrinsic matrix)

**Objective:** minimises reprojection error (3D $\rightarrow$ 2D)

$$\begin{aligned}x_{img} &= P X_{world} \\P &= K[R|t]\end{aligned}$$



Reconstruction (side)



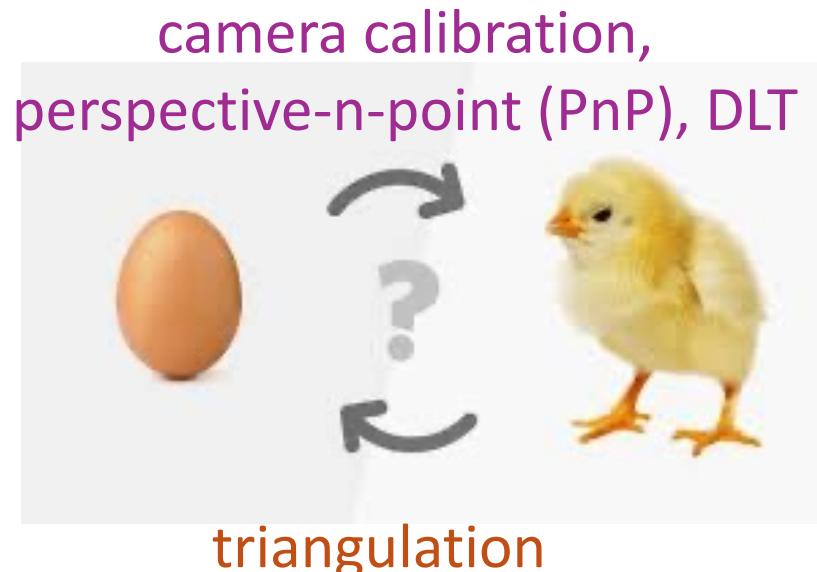
# Structure From Motion

- What is structure from motion?

## Output:

- 1) 3D structure: captured scene (3D location for 2D pixel)
- 2) camera motion: camera parameters (intrinsic and extrinsic matrix)

3D structure

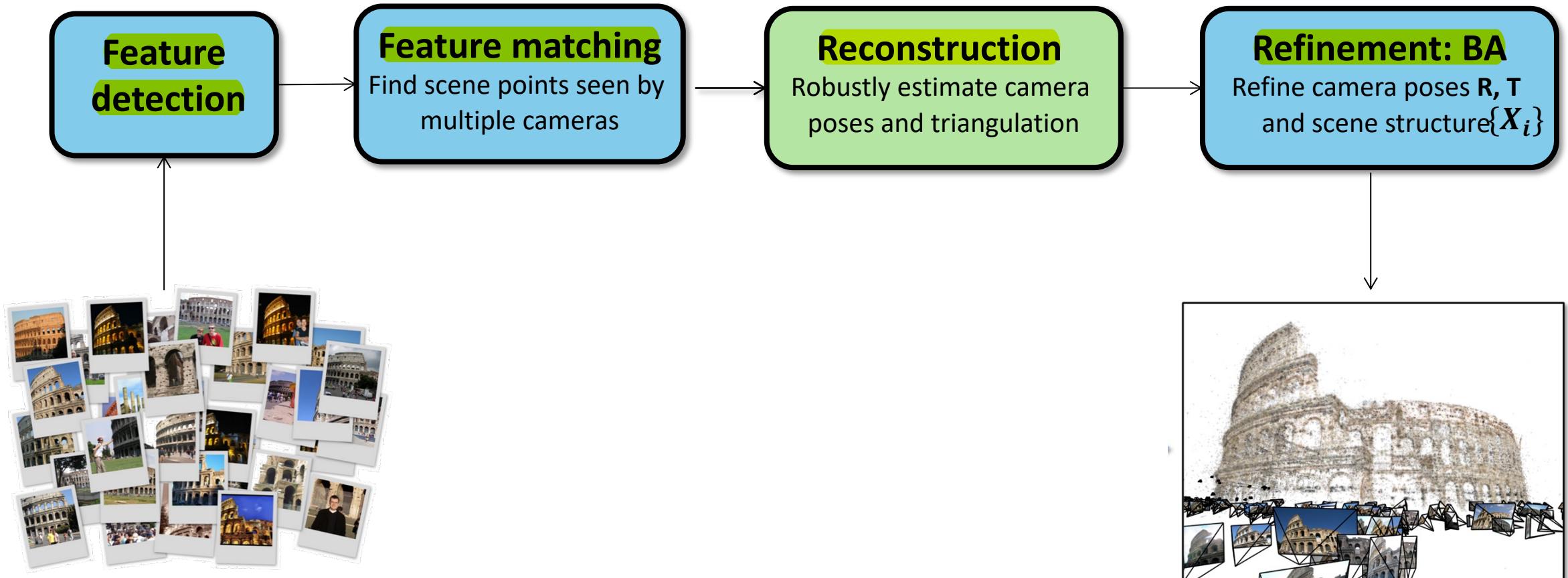


$$x_{img} = P X_{world}$$
$$P = K[R|t]$$

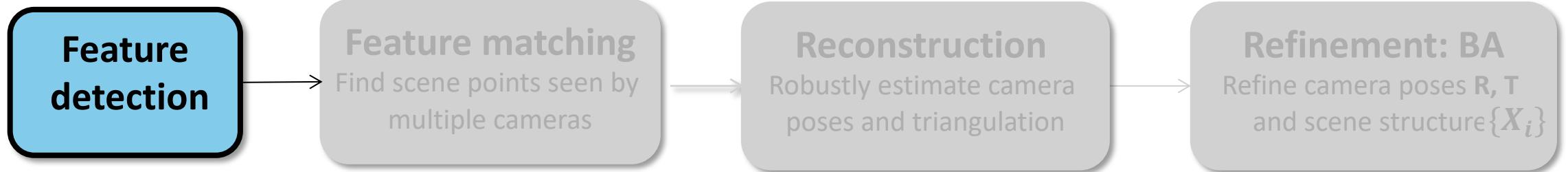
Camera parameters

# Structure From Motion

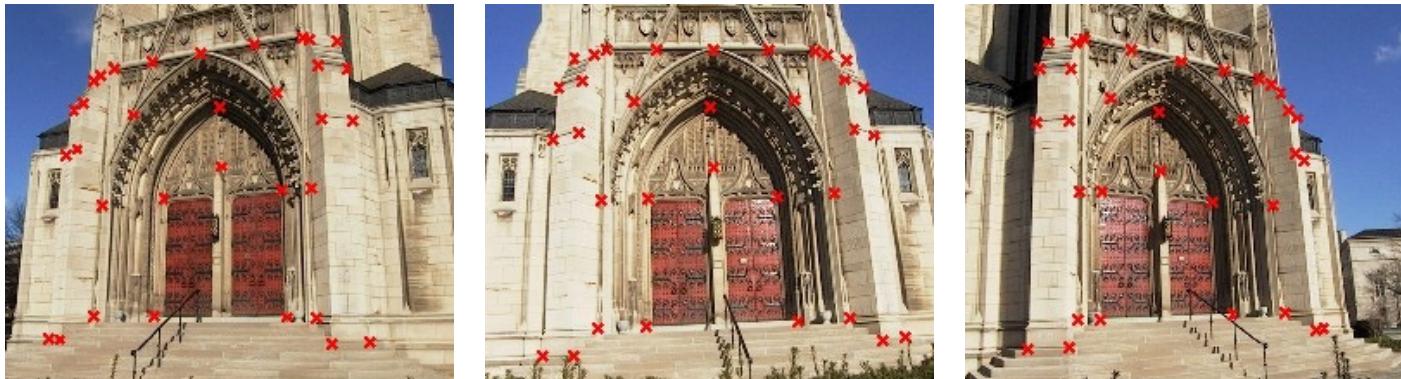
- Core Steps of SfM



# Structure From Motion: 1) Feature Detection

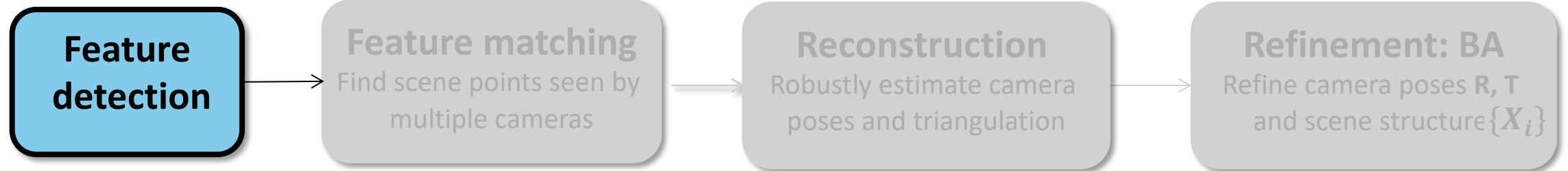


- Identifying **points of interest** within images that can be **tracked across a series**



- SIFT (Scale-Invariant Feature Transform)
- SURF (Speeded Up Robust Features): to scale and rotation changes
- ORB (Oriented FAST and Rotated BRIEF)

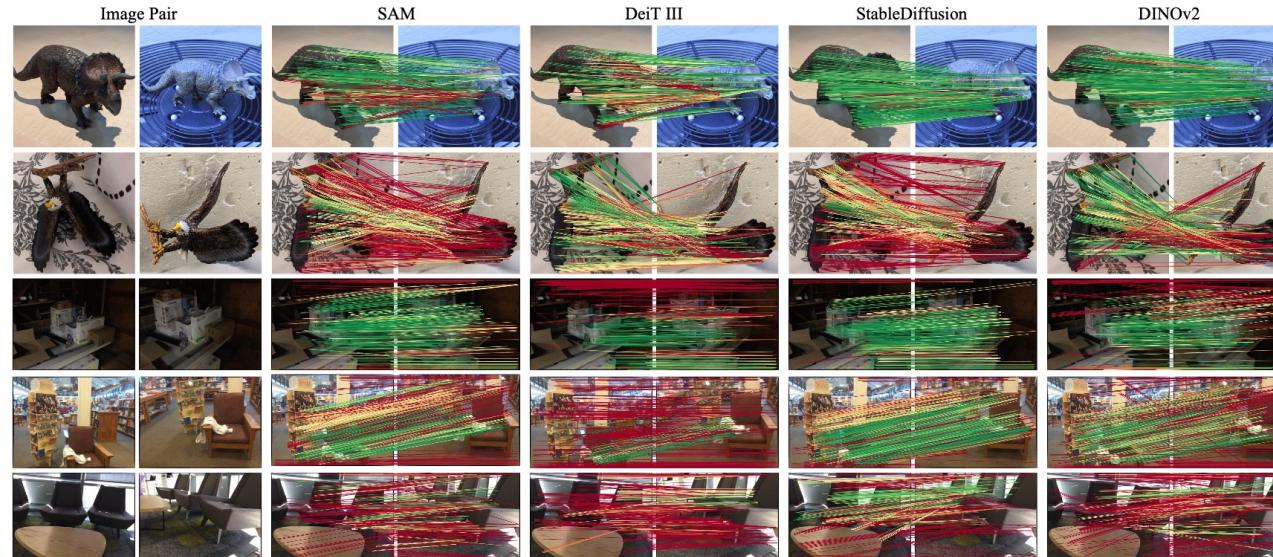
# Structure From Motion: 1) Feature Detection



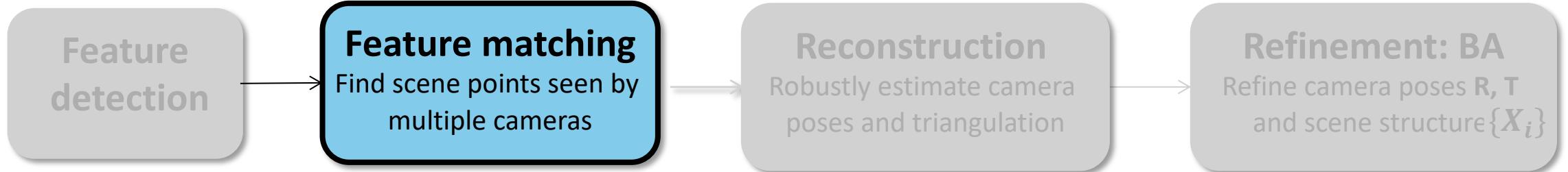
- Identifying points of interest within images that can be tracked across a series

*Probing the 3D Awareness of Visual Foundation Models. In CVPR 2024*

Deep features

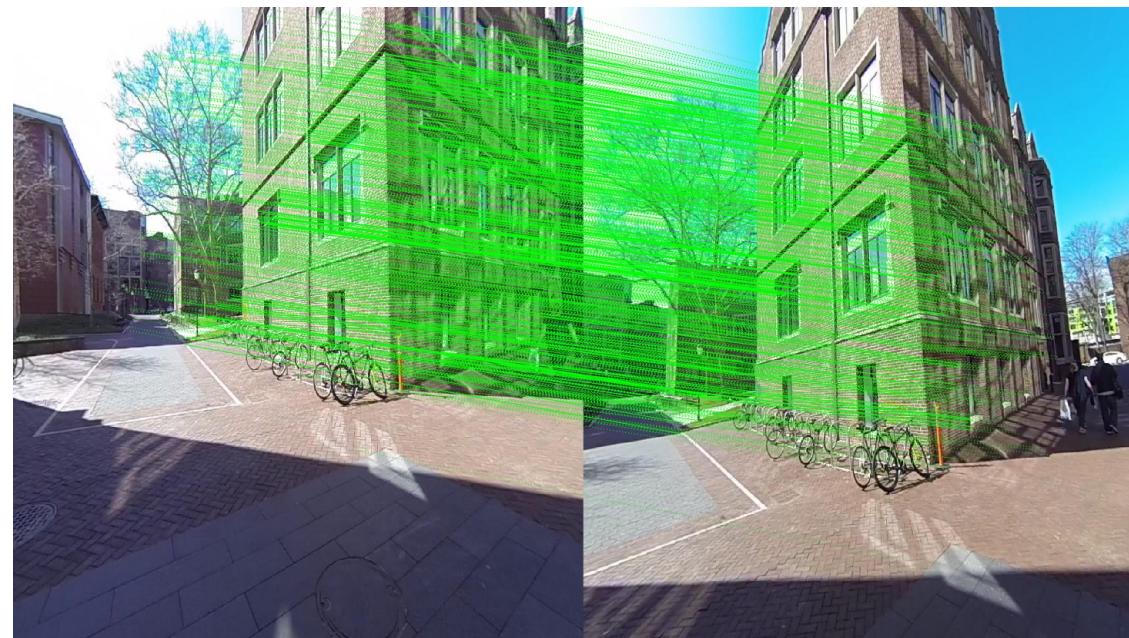


# Structure From Motion: 2) Feature Matching

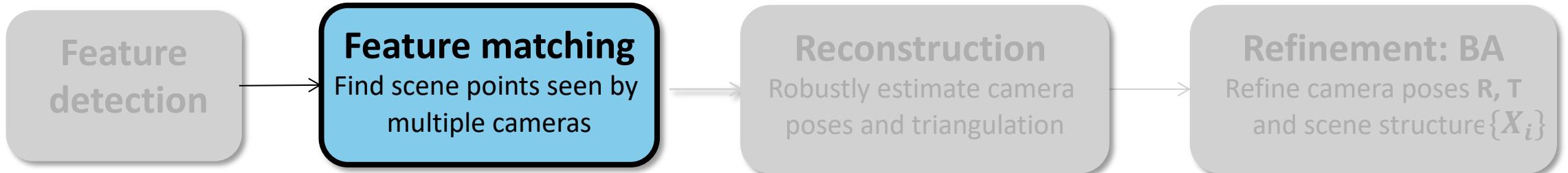


- Establishing **correspondences** between **detected features** across **different images**

Euclidean Distance



# Structure From Motion: 2) Feature Matching

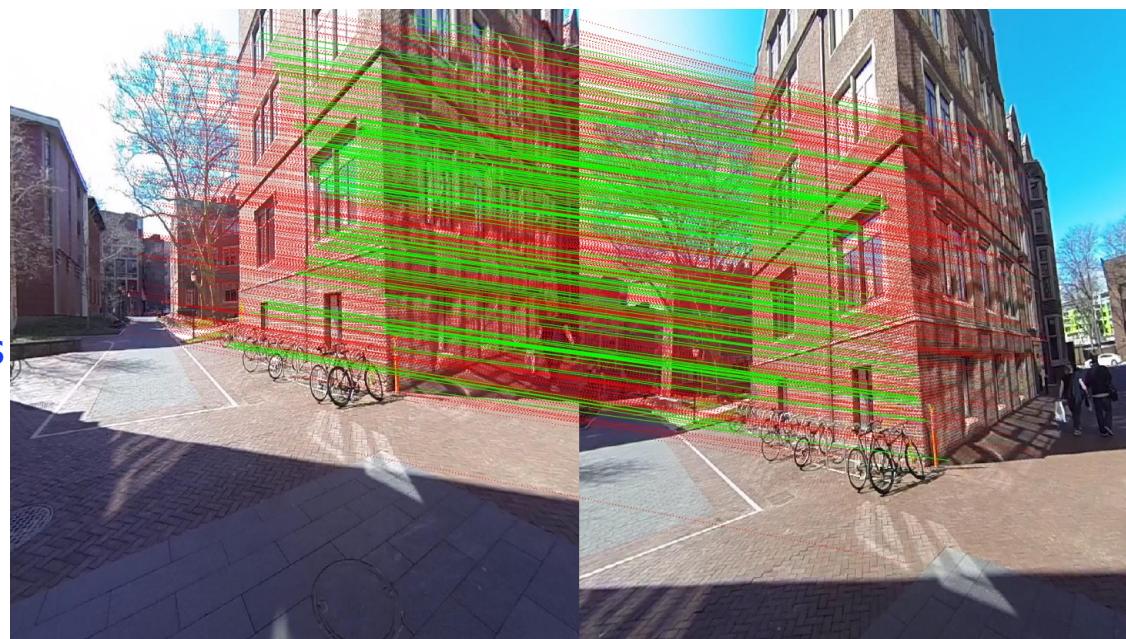


- **RANSAC algorithm** to remove outliers

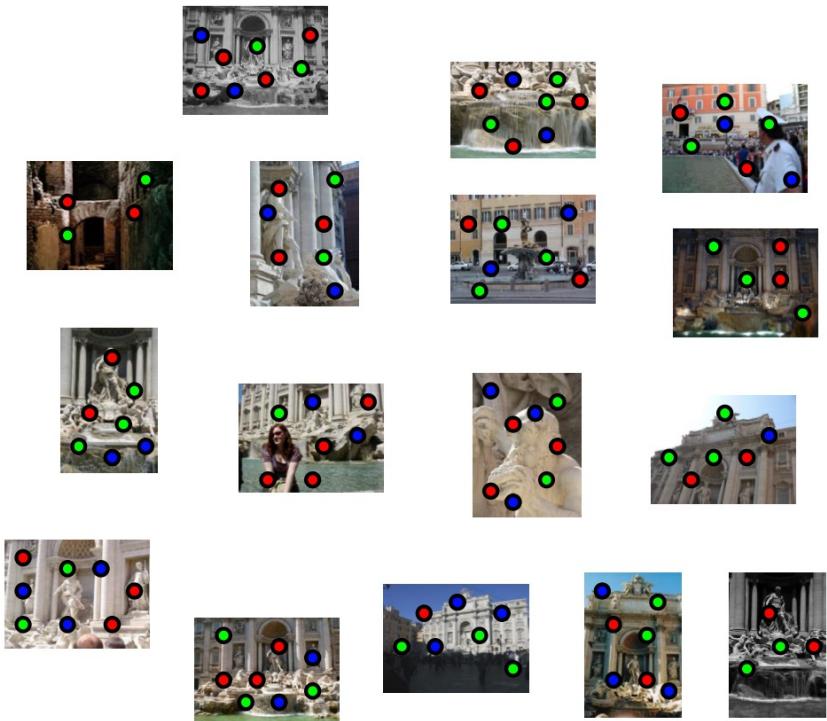
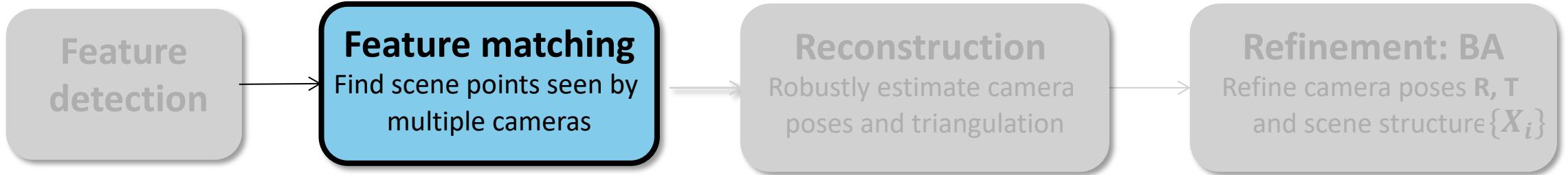
Fundamental Matrix

iteratively selecting a  
subset of the data **inliers**

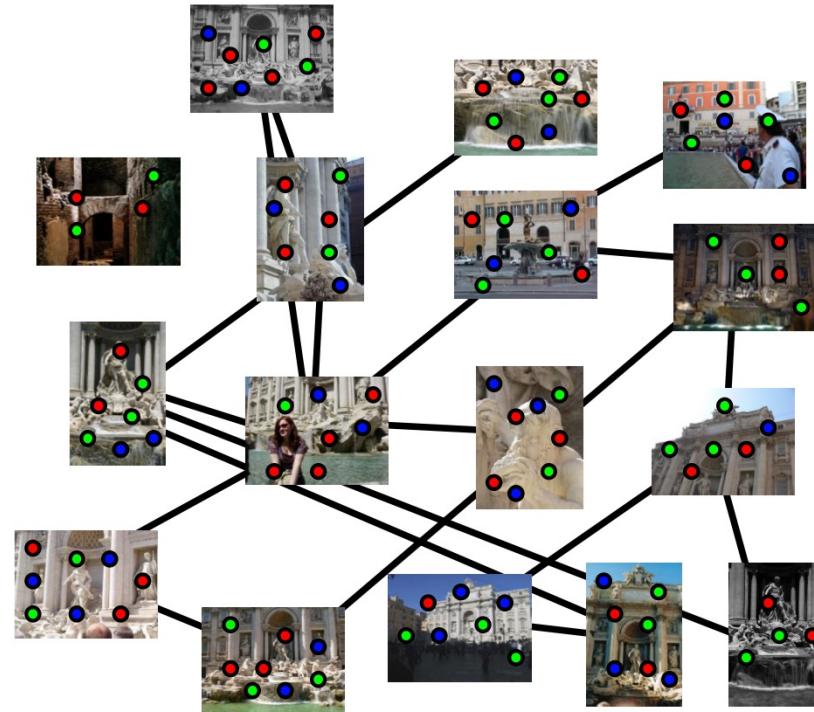
$$x_{img}^{iT} F x_{img} = 0$$



# Structure From Motion: 2) Feature Matching

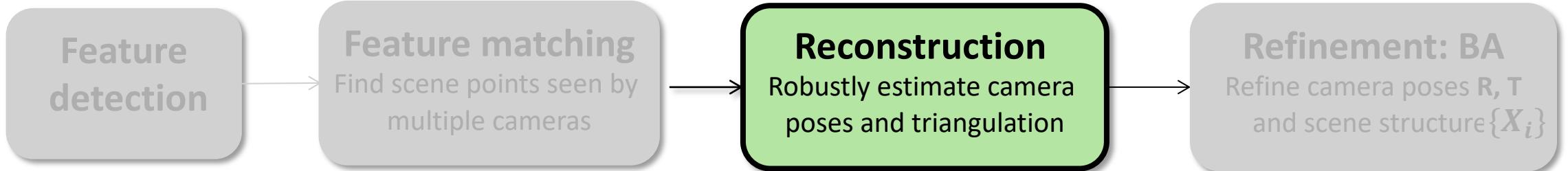


matching

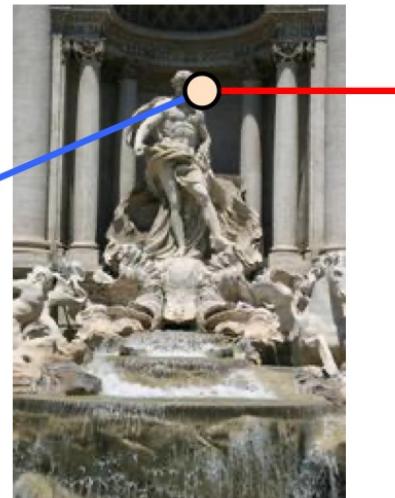
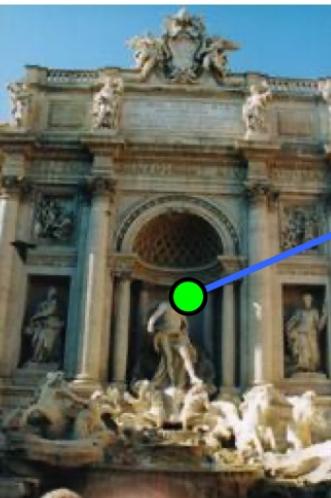


Detect features using SIFT [Lowe, IJCV 2004]

# Structure From Motion: 3) Reconstruction



- Incremental SfM



## Epipolar Geometry

- i) Fundamental Matrix

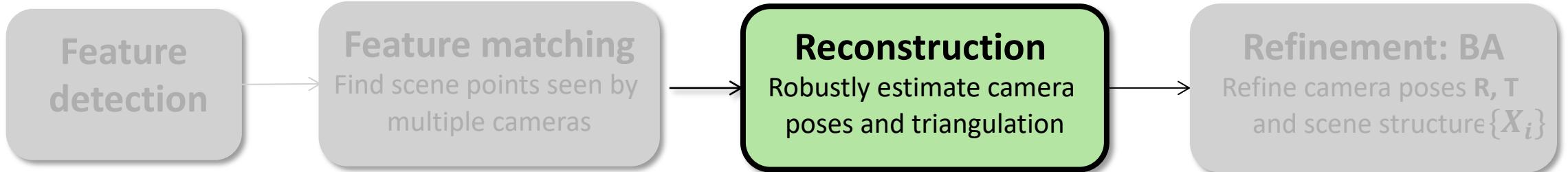
$$\begin{aligned} \mathbf{x}'_{img}^T \mathbf{F} \mathbf{x}_{img} &= 0 \\ \mathbf{F} &= \mathbf{K}'^{-T} \mathbf{E} \mathbf{K}^{-1} \end{aligned}$$

- ii) Essential Matrix

$$\begin{aligned} \mathbf{x}_c'^T \mathbf{E} \mathbf{x}_c &= 0 \\ \mathbf{E} &= [\mathbf{t}]_x \mathbf{R} \end{aligned}$$

- iii) Triangulation: Initial 3D points

# Structure From Motion: 3) Reconstruction



- Incremental SfM: adding a new image

**Reference**



Image 1

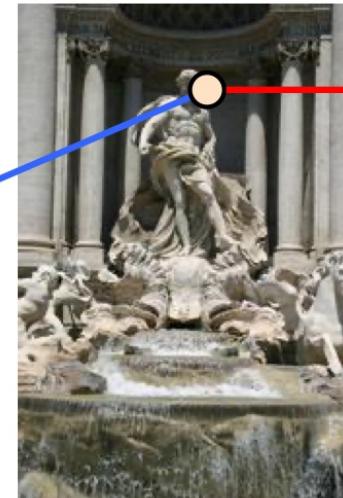


Image 2

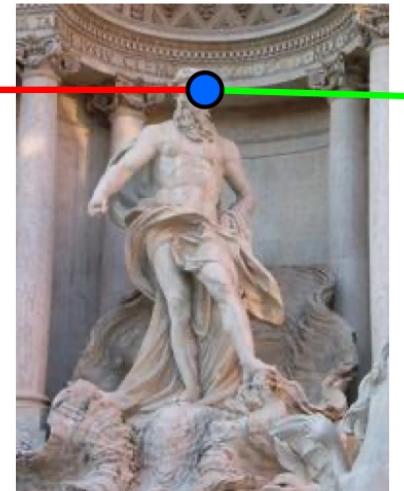


Image 3

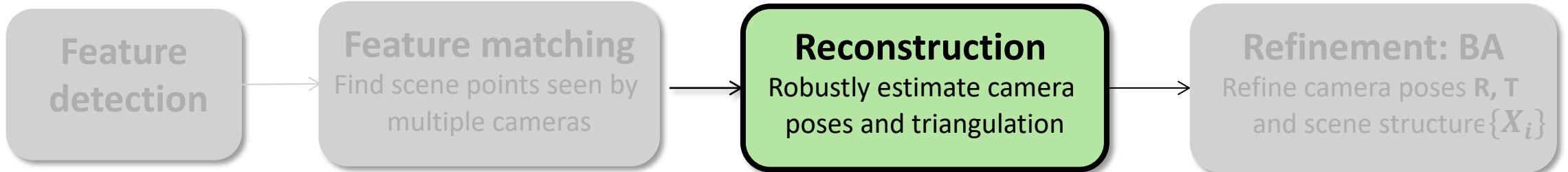
Add new image into existing pair

$$x_{img} = P X_{world}$$
$$P = K[R|t]$$

**Perspective-n-Point (PnP)**

Camera pose for new image

# Structure From Motion: 3) Reconstruction



- Incremental SfM: adding a new image

Reference



Image 1



Image 2

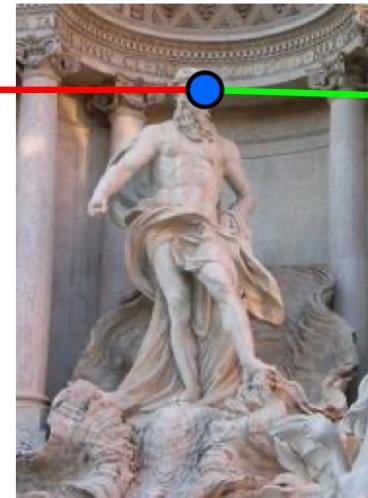


Image 3

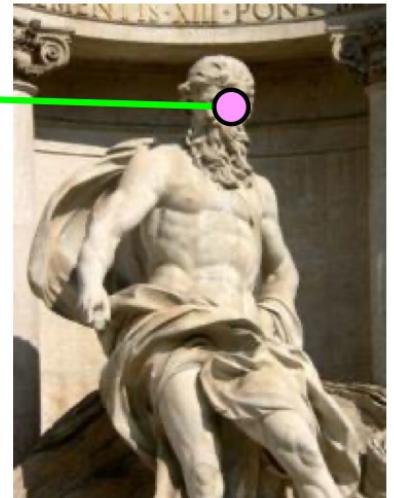
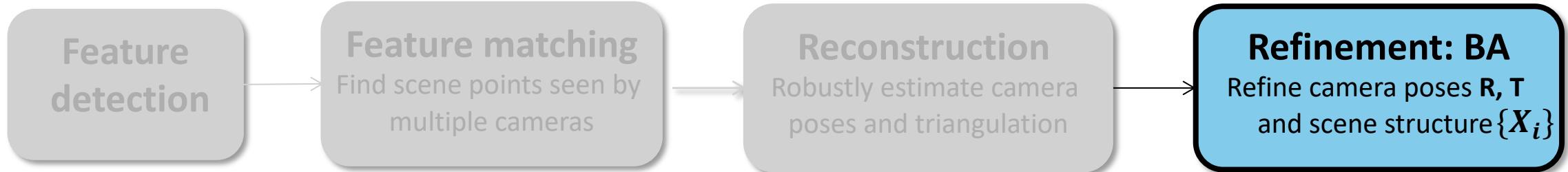


Image 4

# Structure From Motion: 4) Refinement



- Bundle Adjustment

**Input:** Initial 3D structure and camera poses

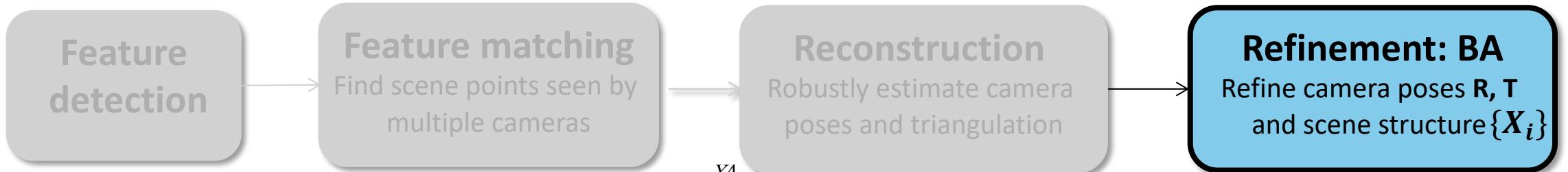
intrinsics mat. K is known for each cam pose

**Goal:** Refine both properties extrinsics matrice: R and t

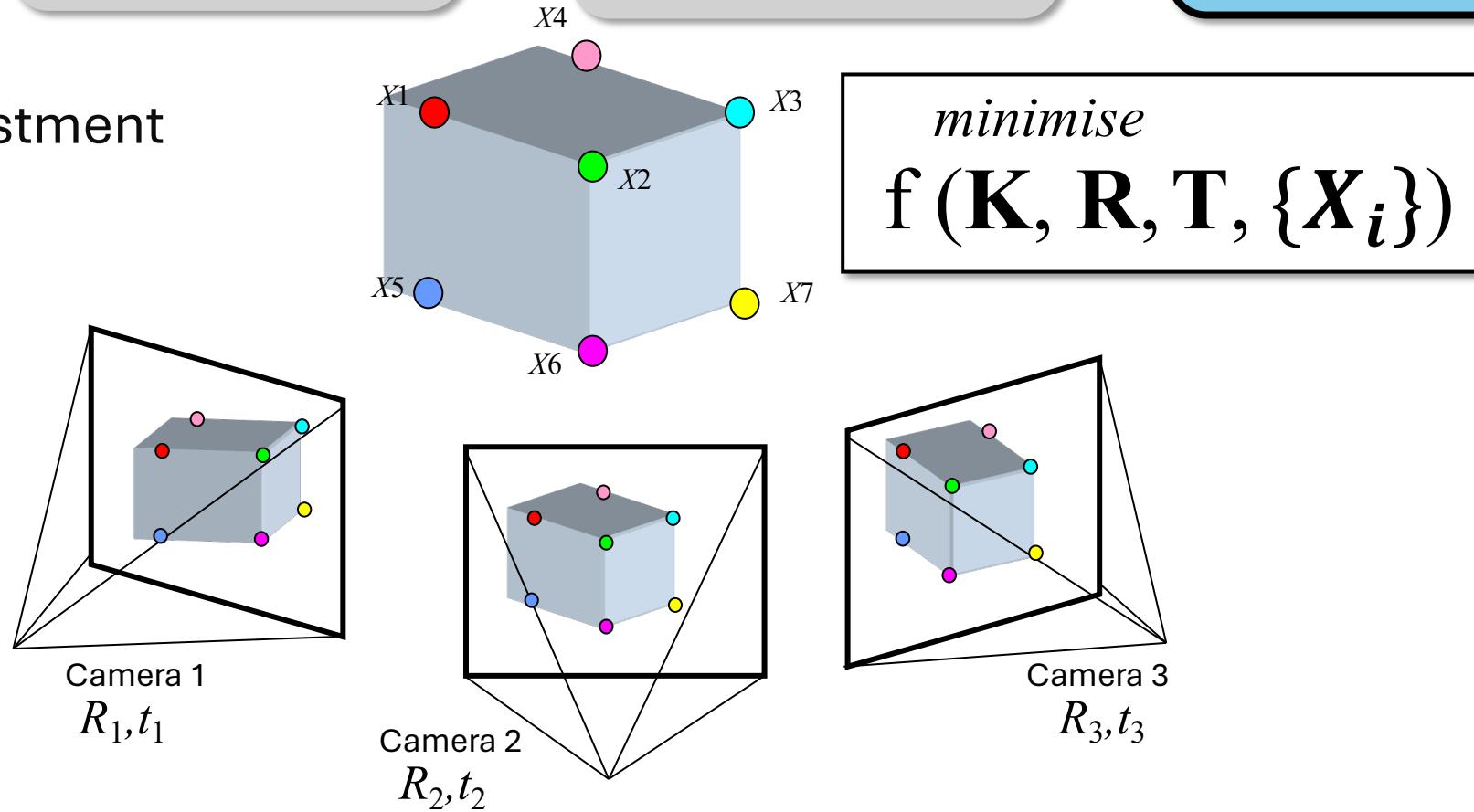
**Objective:** minimises reprojection error (3D → 2D)

$$\begin{aligned}x_{img} &= P X_{world} \\P &= K[R|t]\end{aligned}$$

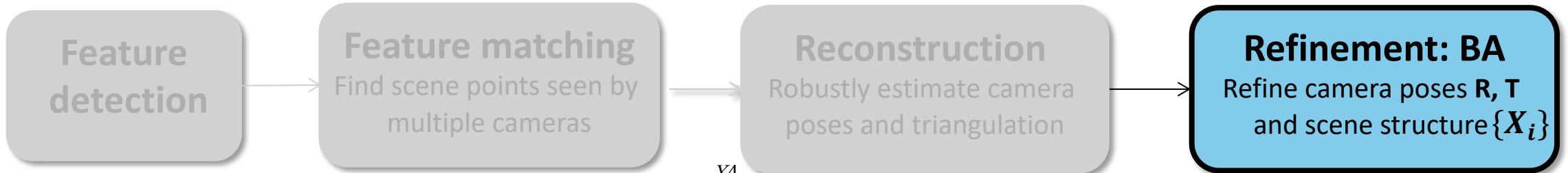
# Structure From Motion: 4) Refinement



- Bundle Adjustment

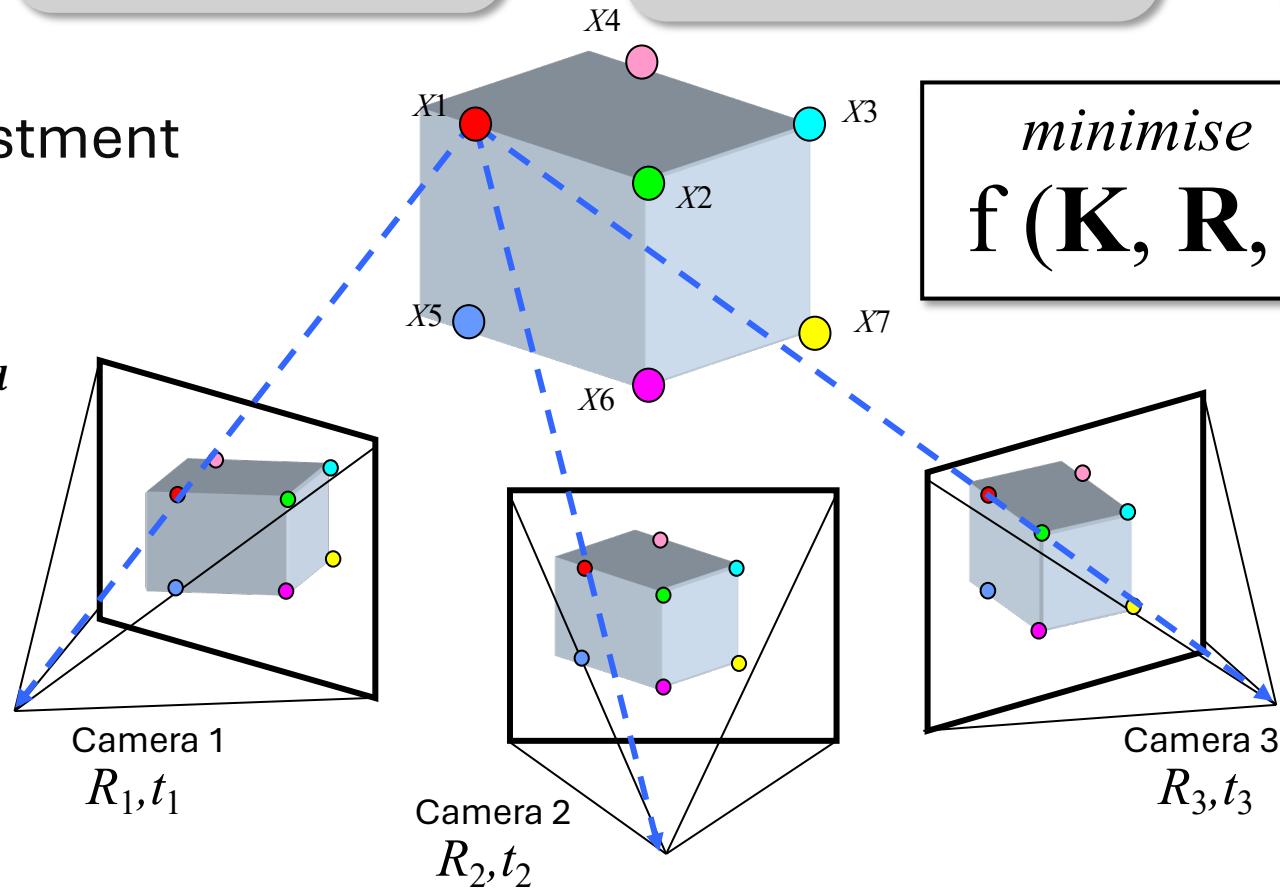


# Structure From Motion: 4) Refinement

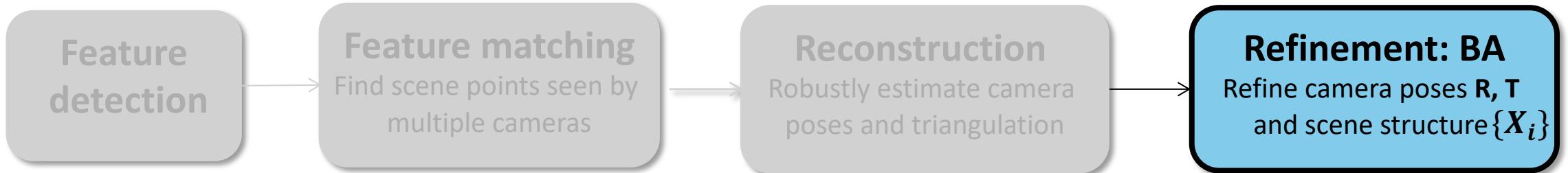


- Bundle Adjustment

$$x_{img} = P X_{world}$$
$$P = K[R|t]$$



# Structure From Motion: 4) Refinement



- **Bundle Adjustment**

Given:  $m$  images of  $n$  fixed 3D points

$x_{ij}$ : j-th point on i-th pic (cam)

$m$  cams

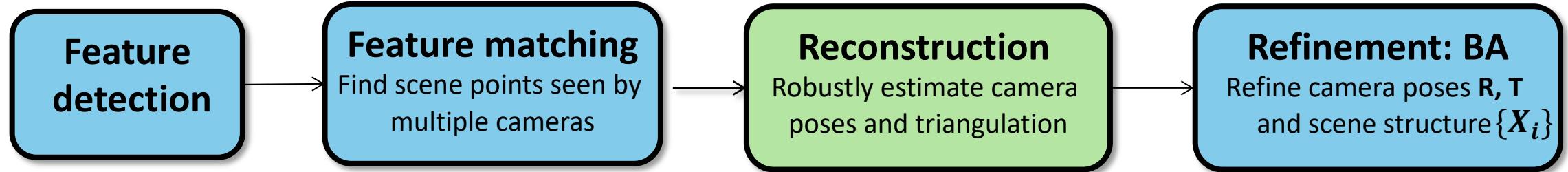
$n$  3D points

$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

for each cam

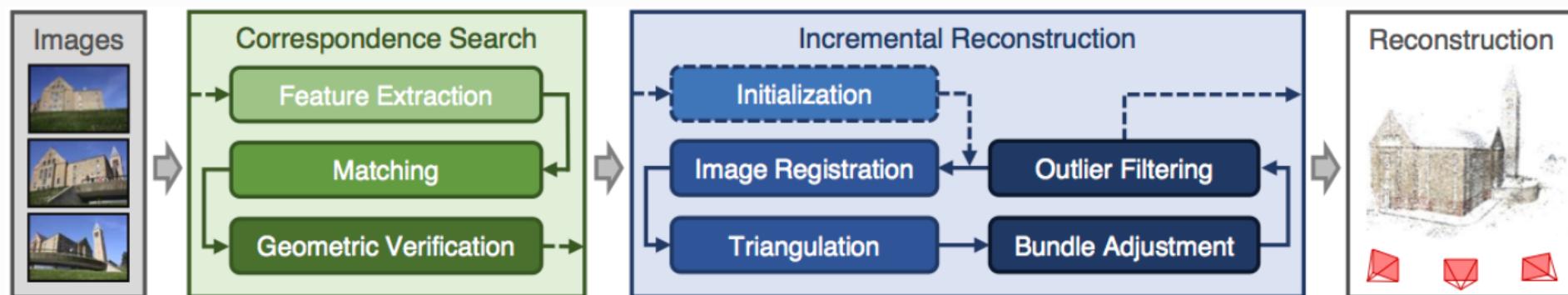
Problem: estimate  $m$  projection matrices  $\mathbf{P}_i$  and  $n$  3D points  $\mathbf{X}_j$  from the  $mn$  corresponding points  $\mathbf{x}_{ij}$

# Structure From Motion: COLAMP



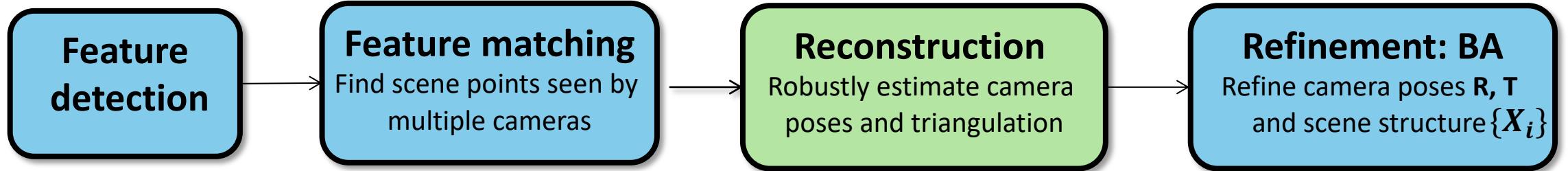
<https://colmap.github.io/tutorial.html#structure-from-motion>

## Structure-from-Motion

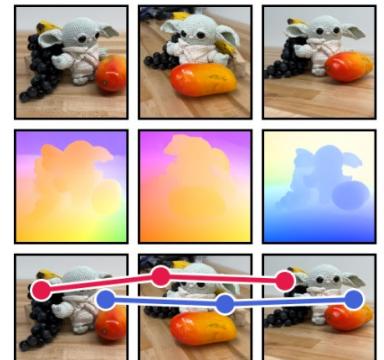


COLMAP's incremental Structure-from-Motion pipeline.

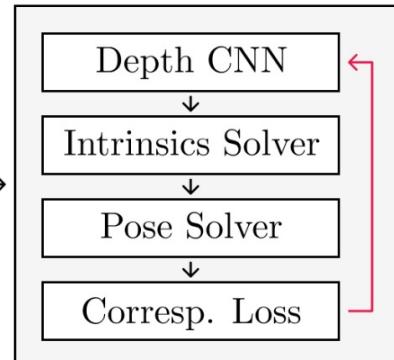
# Structure From Motion: FlowMap



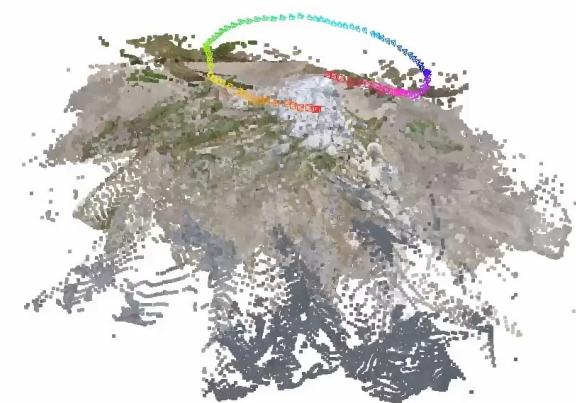
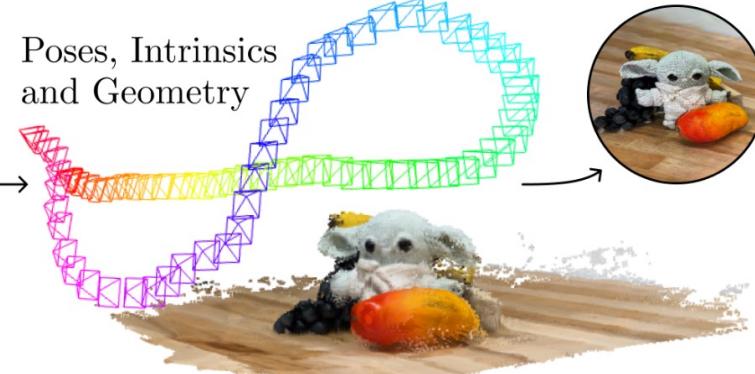
Video and Off-the-Shelf Correspondences



FlowMap Optimization  
via **Gradient Descent**

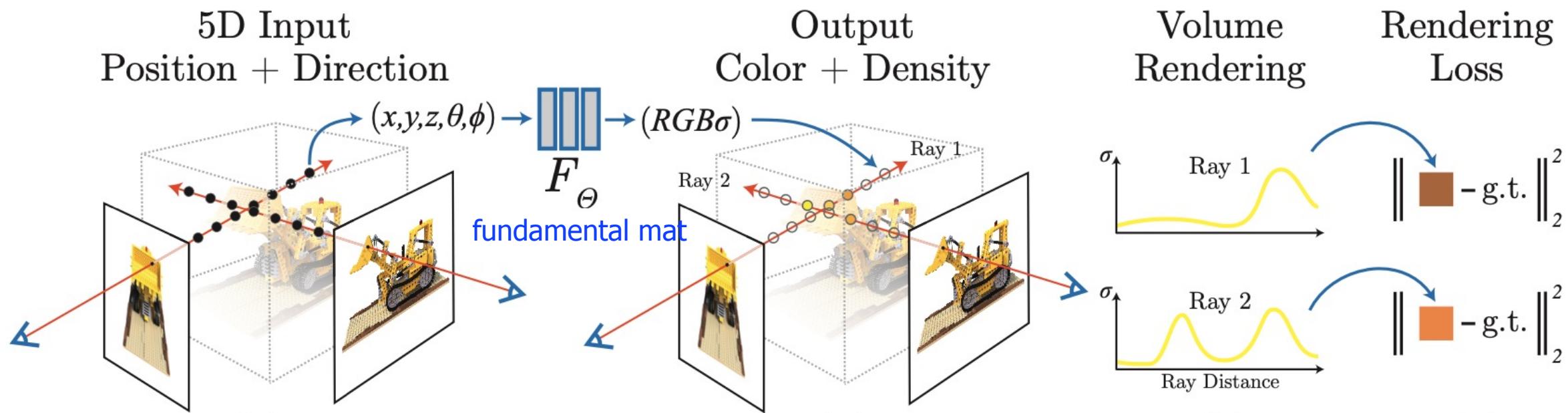


Downstream Task:  
Gaussian Splatting



# Neural Radiance Fields (NeRFs)

- NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis  
*(representing 3D scenes using neural networks/MLPs)*



# Neural Radiance Fields (NeRFs)

- NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis  
*(representing 3D scenes using neural networks/MLPs)*

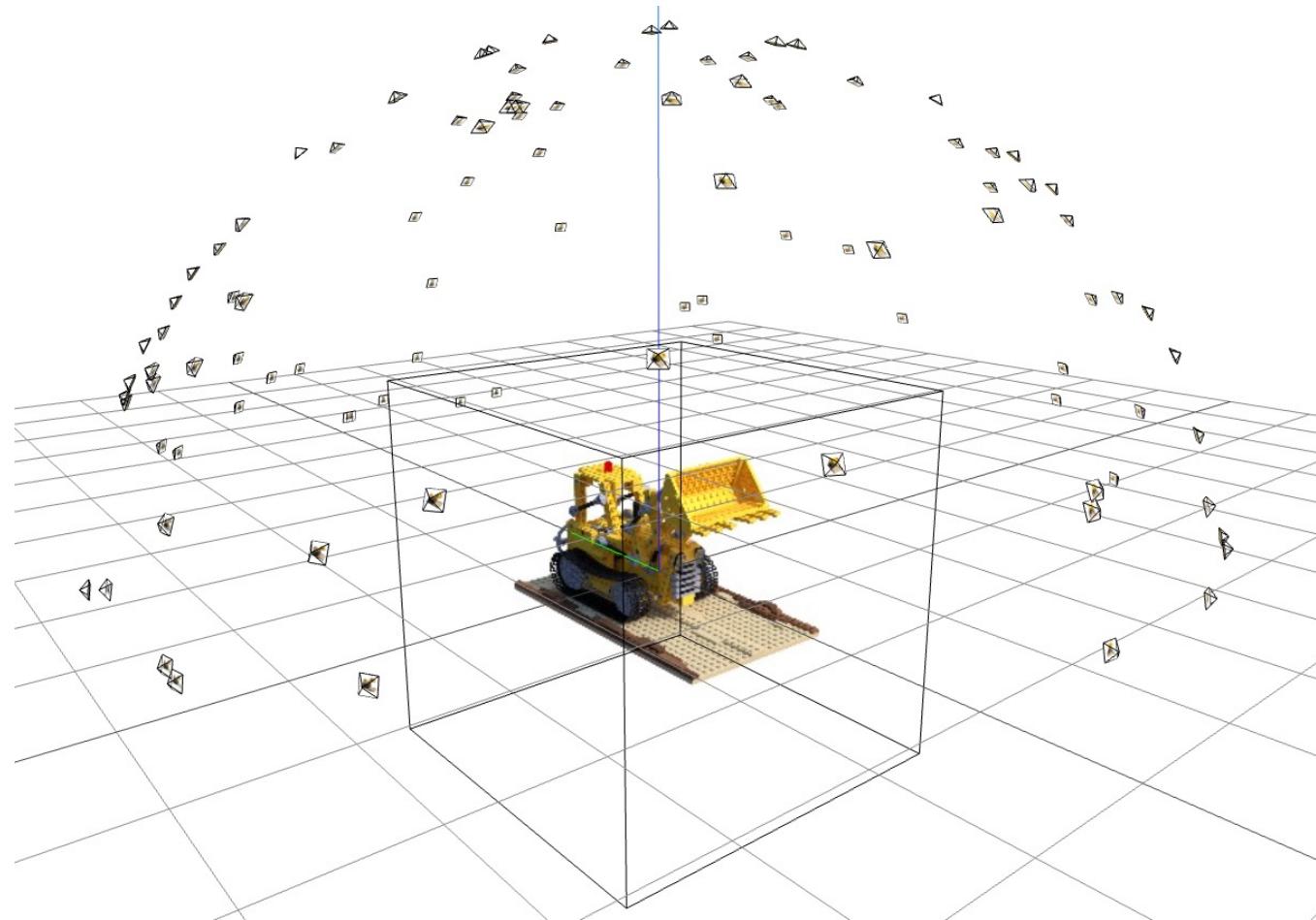
**Input:** multiple camera images (or video) + associated camera poses

**Output:** Neural Radiance Field (MLP weights)

**Objective:** minimises photometric error  
(rendered image vs ground-truth image)

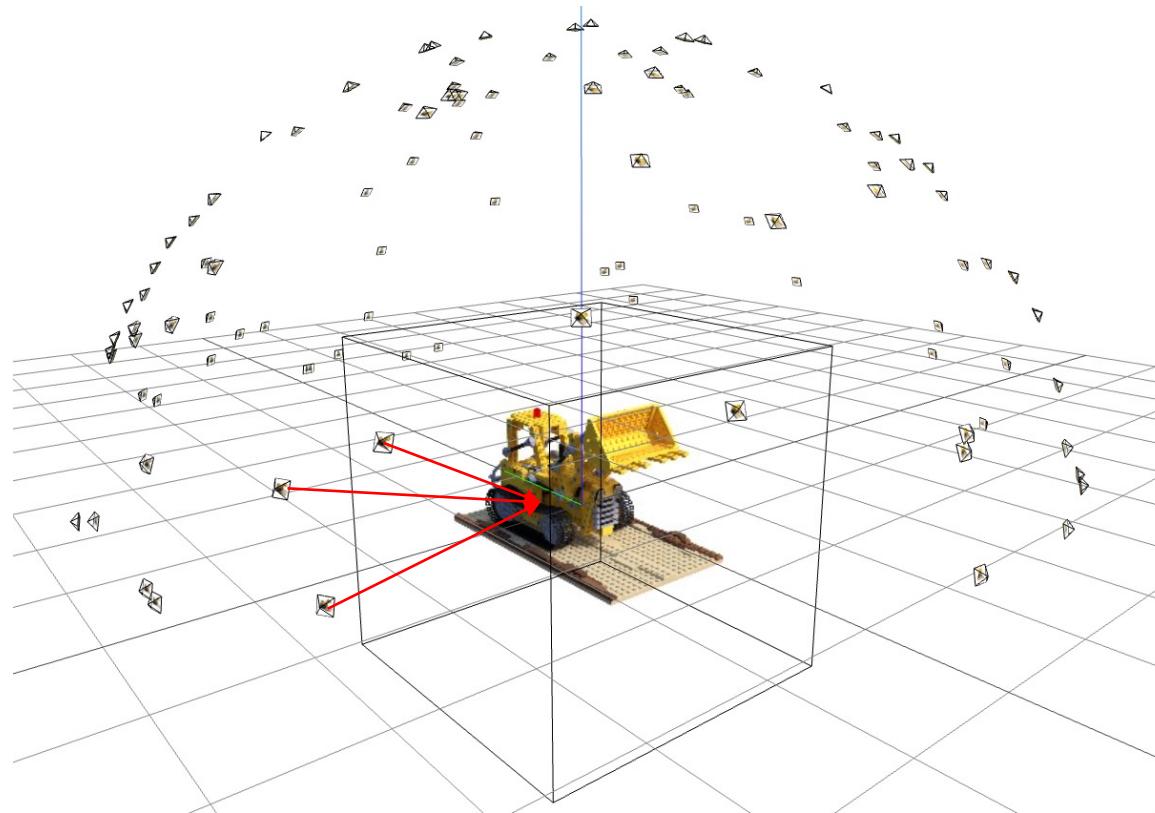
# Neural Radiance Fields (NeRFs)

- Step 1: let take photos



# Neural Radiance Fields (NeRFs)

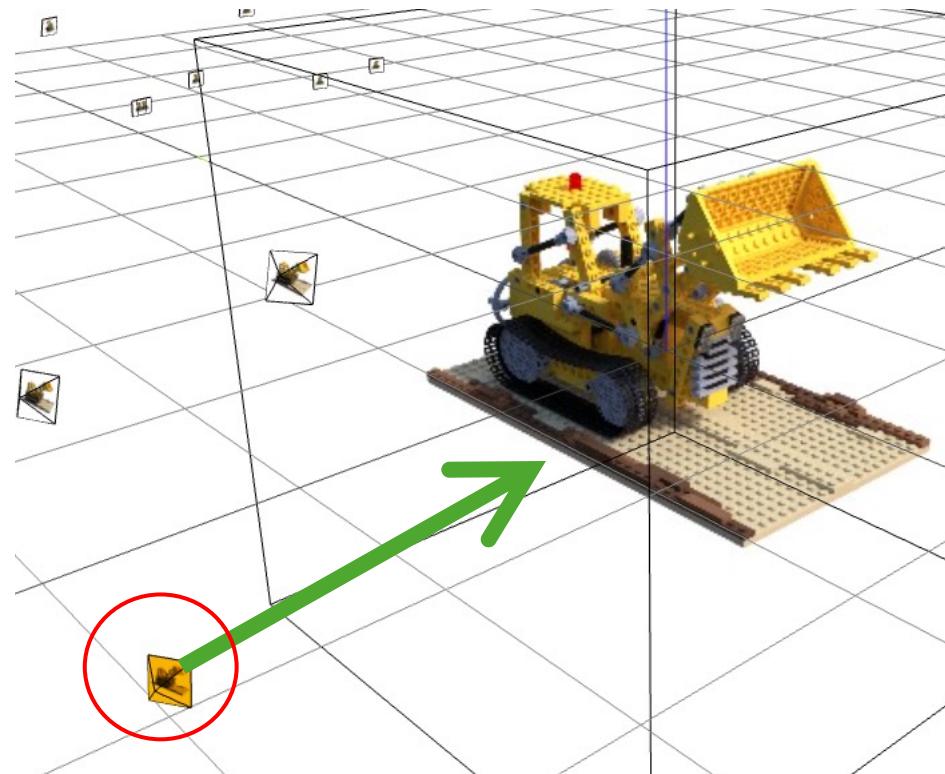
- Step 2: using Colmap (SFM) to calculate camera poses  
*(detecting and matching features across the images)*



# Neural Radiance Fields (NeRFs)

- Step 3: Volumetric formulation of NeRF

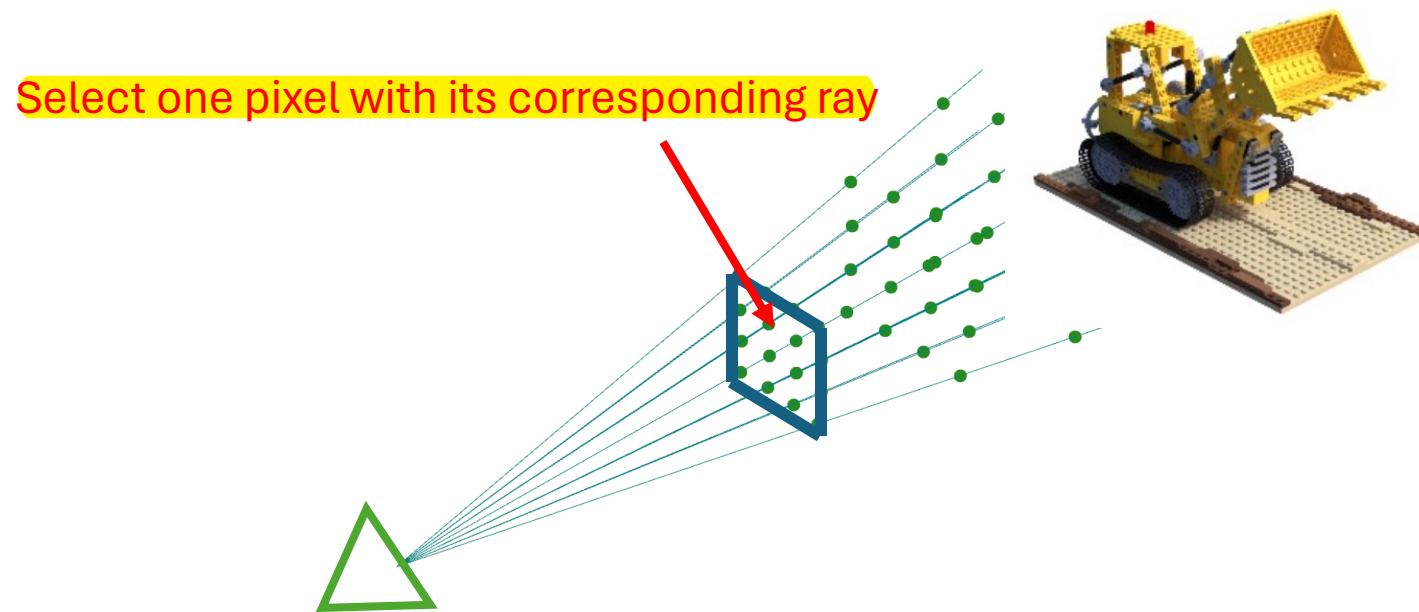
*What will the image/ pixels captured by this camera look like?*



# Neural Radiance Fields (NeRFs)

- Step 3: Volumetric formulation of NeRF

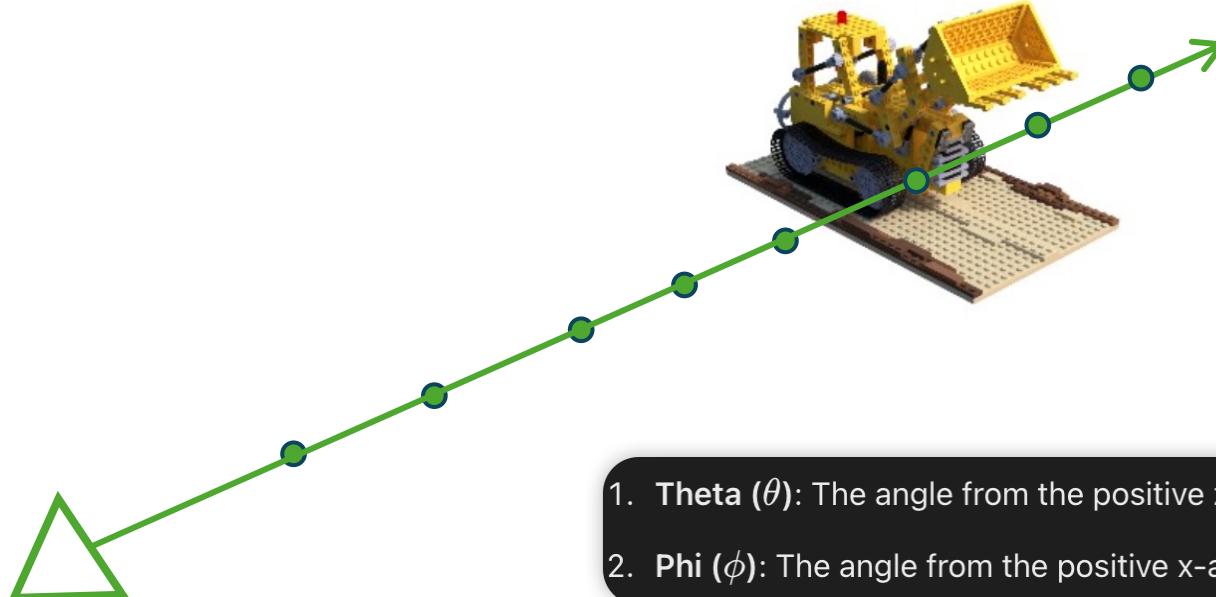
*What will the image/ pixels captured by this camera look like?*



# Neural Radiance Fields (NeRFs)

- Step 3: Volumetric formulation of NeRF

**Ray tracing** for this pixel (straight ray)



1. Theta ( $\theta$ ): The angle from the positive z-axis, also known as the polar angle.
2. Phi ( $\phi$ ): The angle from the positive x-axis in the xy-plane, also known as the azimuthal angle.

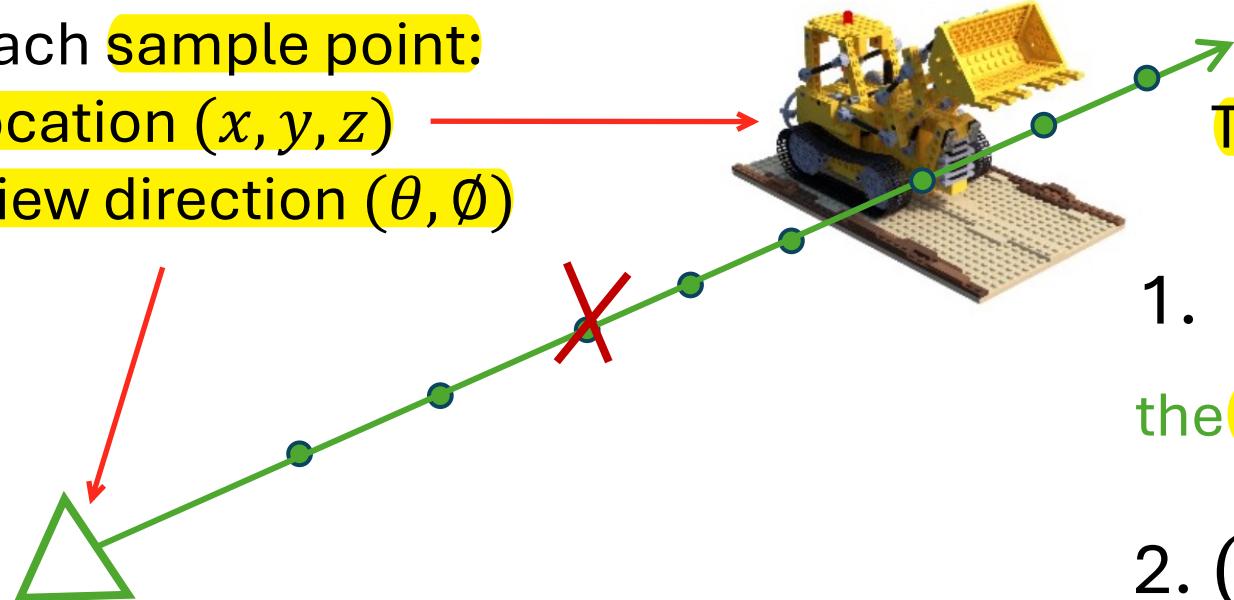
Each sample point, we know its location  $(x, y, z)$  and view direction  $(\theta, \phi)$

# Neural Radiance Fields (NeRFs)

- Step 3: Volumetric formulation of NeRF

**Ray tracing** for this pixel (straight ray)

Each sample point:  
location  $(x, y, z)$   
view direction  $(\theta, \phi)$



Two MLPs:

$$1. (x, y, z) \xrightarrow{f} \sigma$$

the chance the ray hits a particle

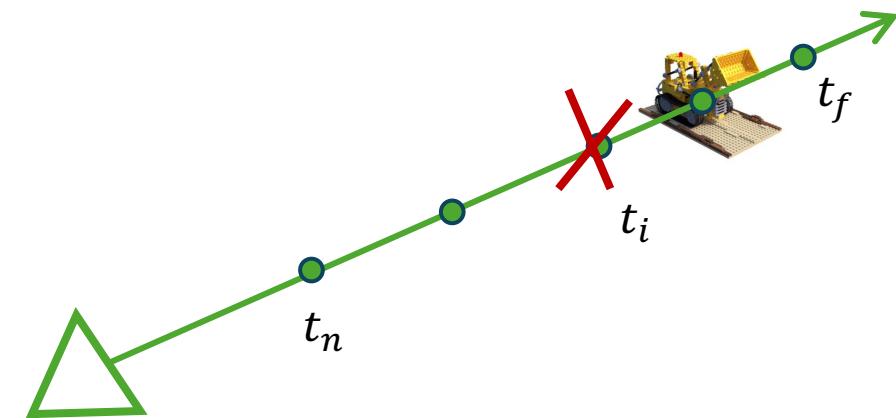
$$2. (x, y, z, \theta, \phi) \xrightarrow{g} \text{color}$$

color is view dependent

# Neural Radiance Fields (NeRFs)

- Step 3: Volumetric formulation of NeRF

**Accumulation** for this ray and get its rendered pixel



$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt ,$$

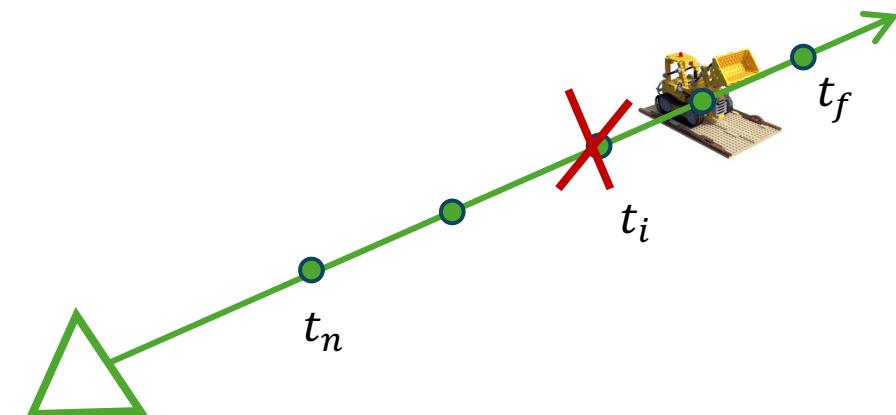
$$T(t) = \exp\left(- \int_{t_n}^t \sigma(\mathbf{r}(s)) ds\right)$$

T: how much the light is blocked earlier along the ray

# Neural Radiance Fields (NeRFs)

- Step 3: Volumetric formulation of NeRF

**Accumulation** for this ray and get its rendered pixel



$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i$$
$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$$

T: how much the light is blocked earlier along the ray

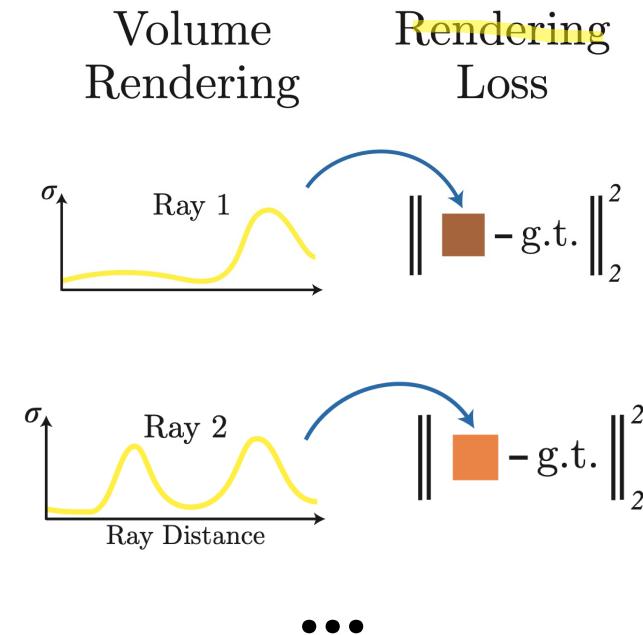
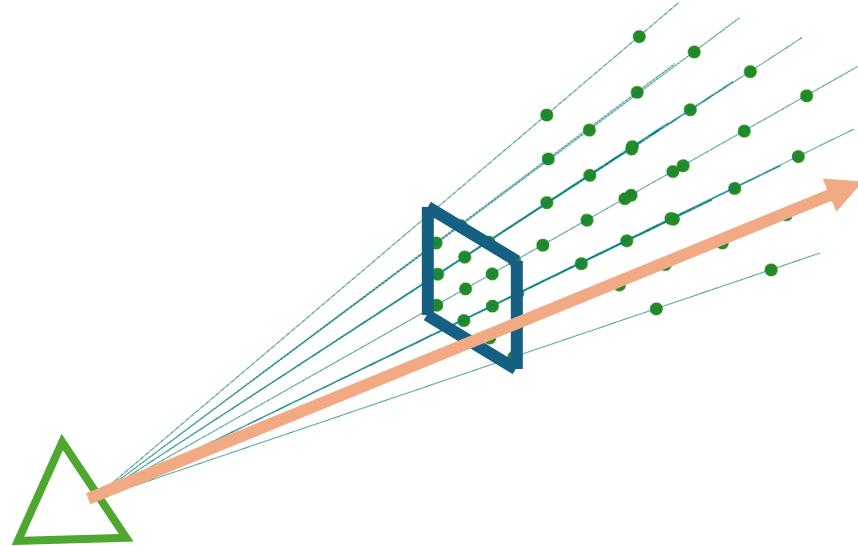
$$\delta_i = t_{i+1} - t_i$$

distance between adjacent point

# Neural Radiance Fields (NeRFs)

- Step 3: Volumetric formulation of NeRF

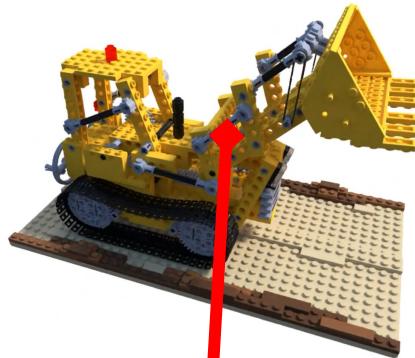
**Accumulation** for this ray and get its rendered pixel



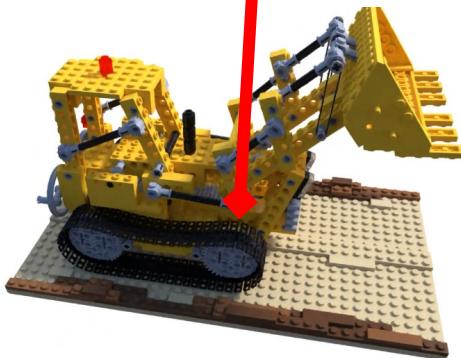
# Neural Radiance Fields (NeRFs)

- Solves correspondences + triangulation implicitly

Camera view A



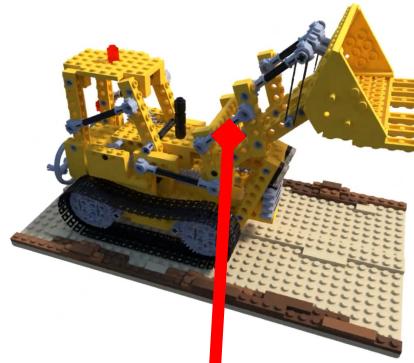
Camera view B



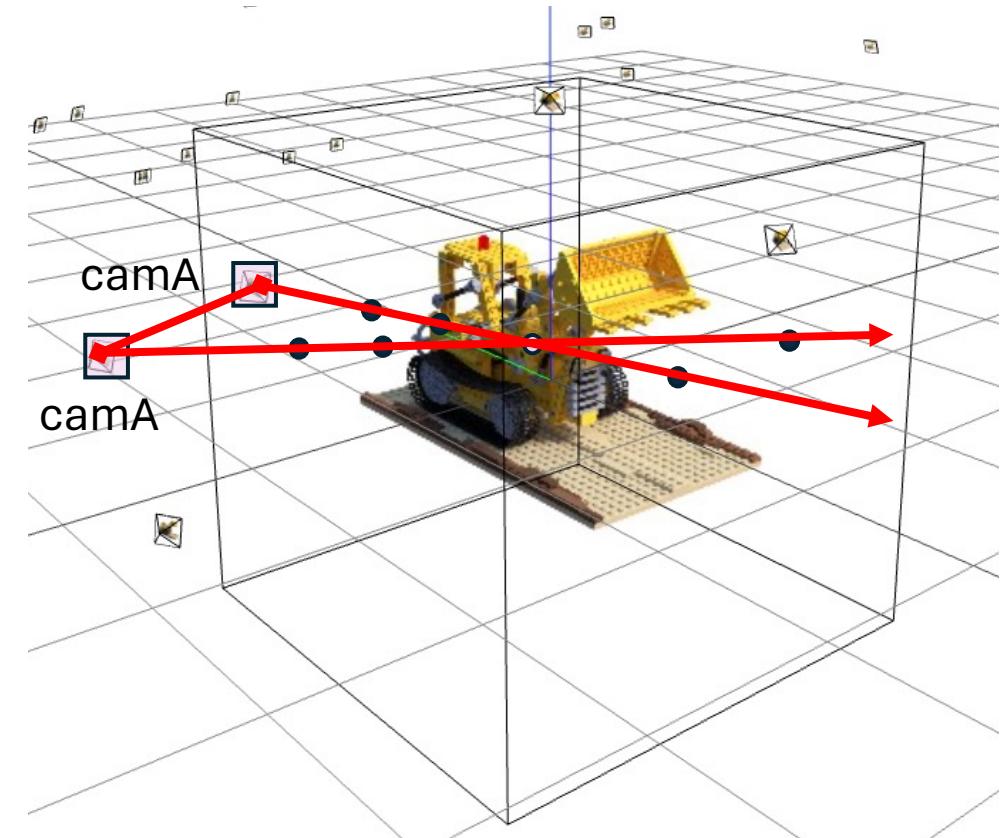
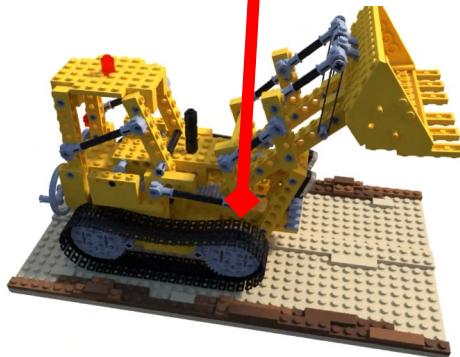
# Neural Radiance Fields (NeRFs)

- Solves **correspondences + triangulation implicitly**

Camera view A

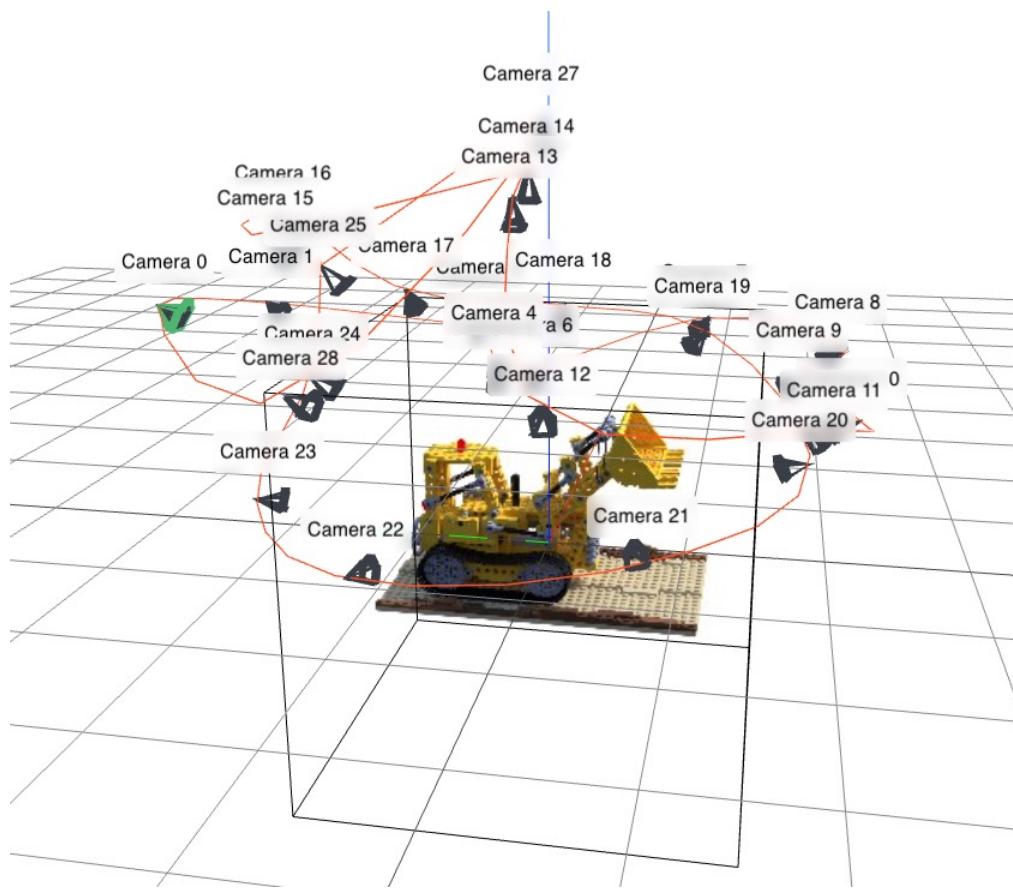


Camera view B



# Neural Radiance Fields (NeRFs)

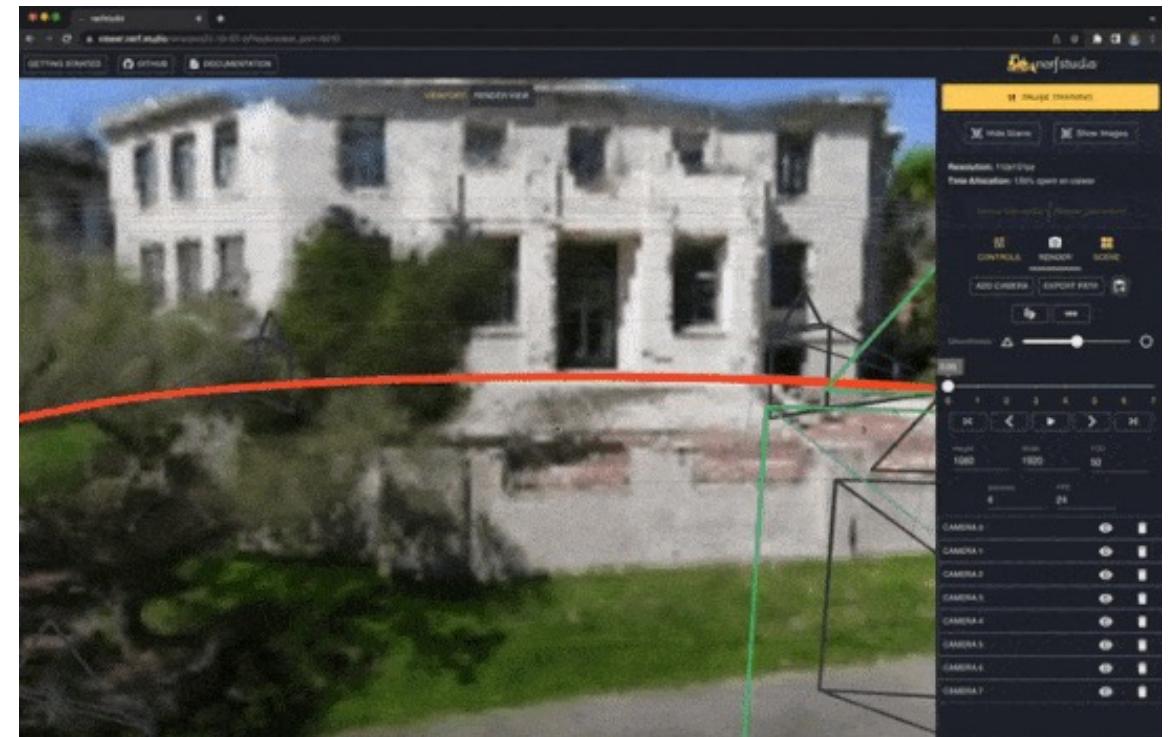
- Novel view synthesis (unseen camera views)



# Neural Radiance Fields (NeRFs)

- More Examples

Building



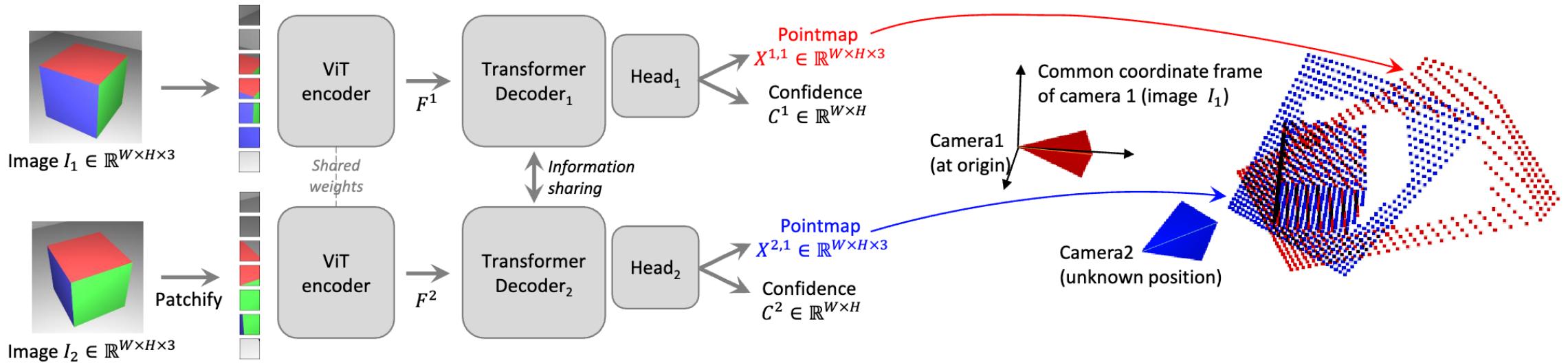
# Neural Radiance Fields (NeRFs)

- More Examples



Human

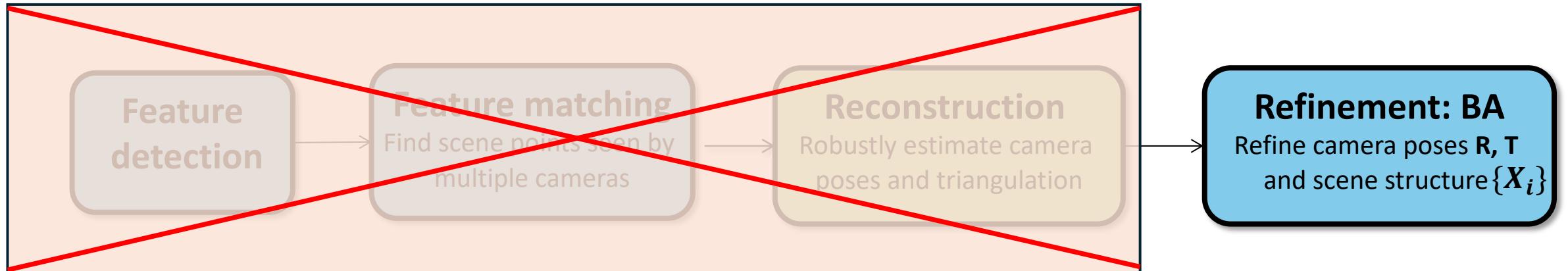




Wang et.al., DUST3R: Geometric 3D Vision Made Easy. In CVPR 2024

# DUST3R: Geometric 3D Vision Made Easy

“Unify all 3D vision tasks”



# DUST3R: Geometric 3D Vision Made Easy

"Unify all 3D vision tasks"



**Input:** multiple camera views (without intrinsic)



**Output:** 3D reconstruction + camera poses

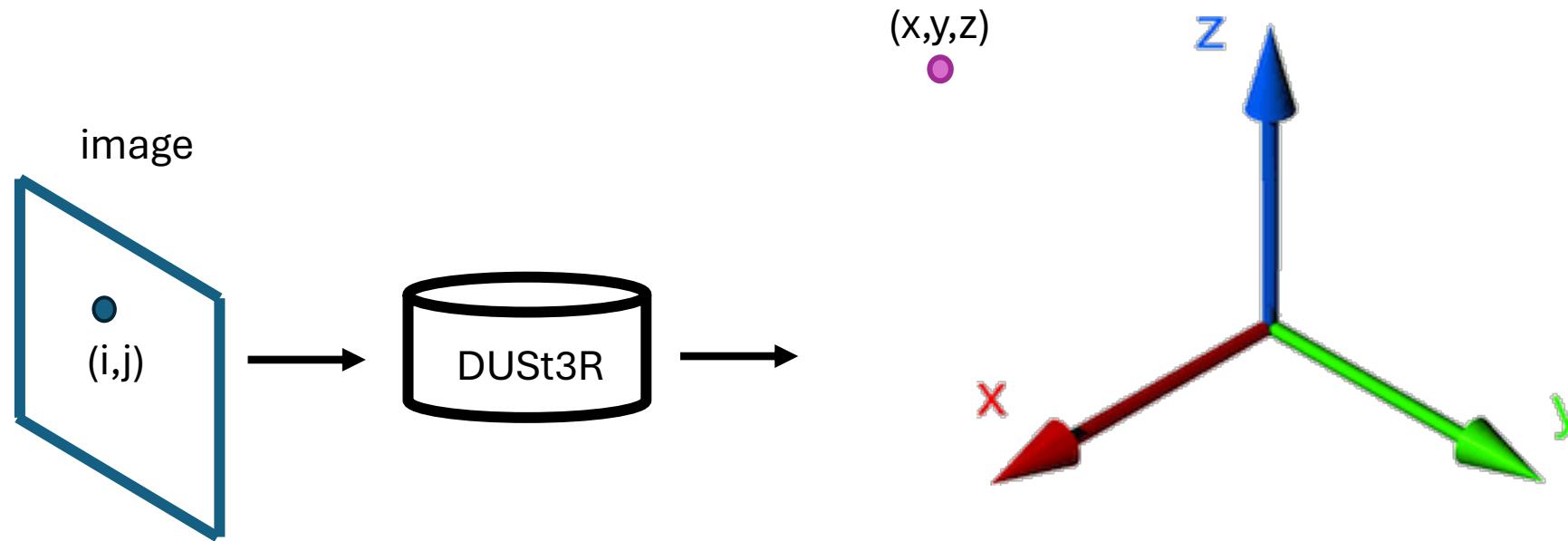


**Objective:** minimises 3D matching loss

DUSt3R: Geometric 3D Vision  
Made Easy

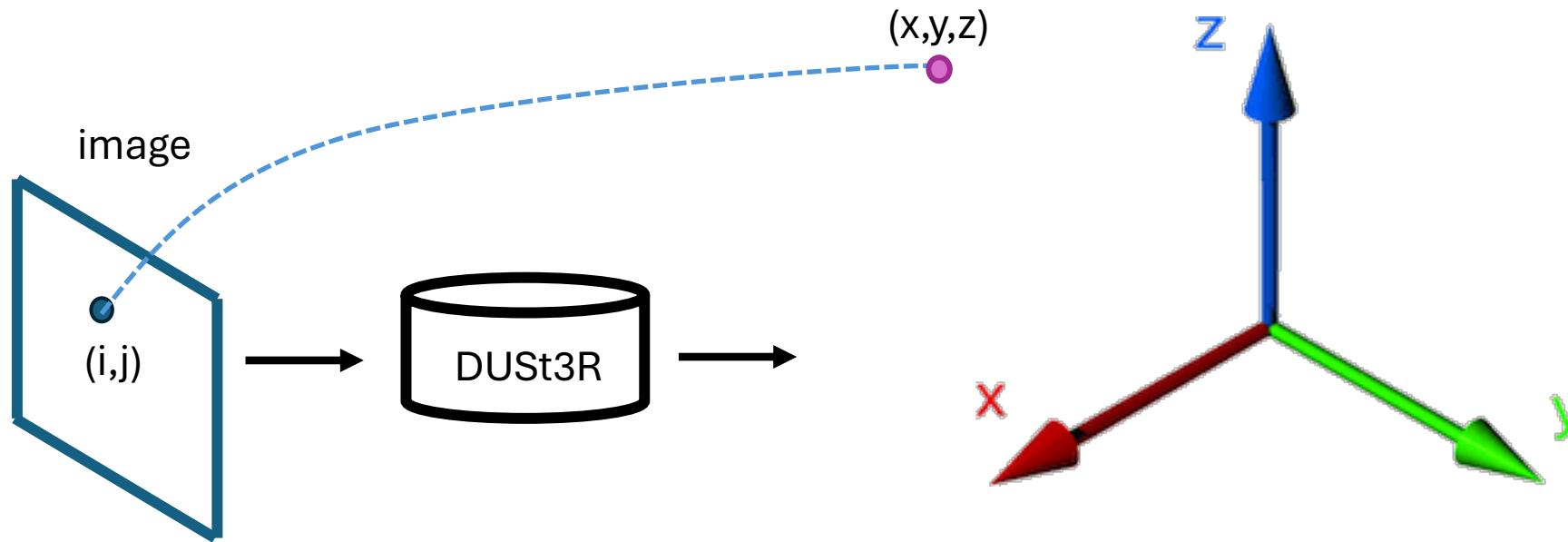
“Unify all 3D vision tasks”

# DUST3R: Geometric 3D Vision Made Easy



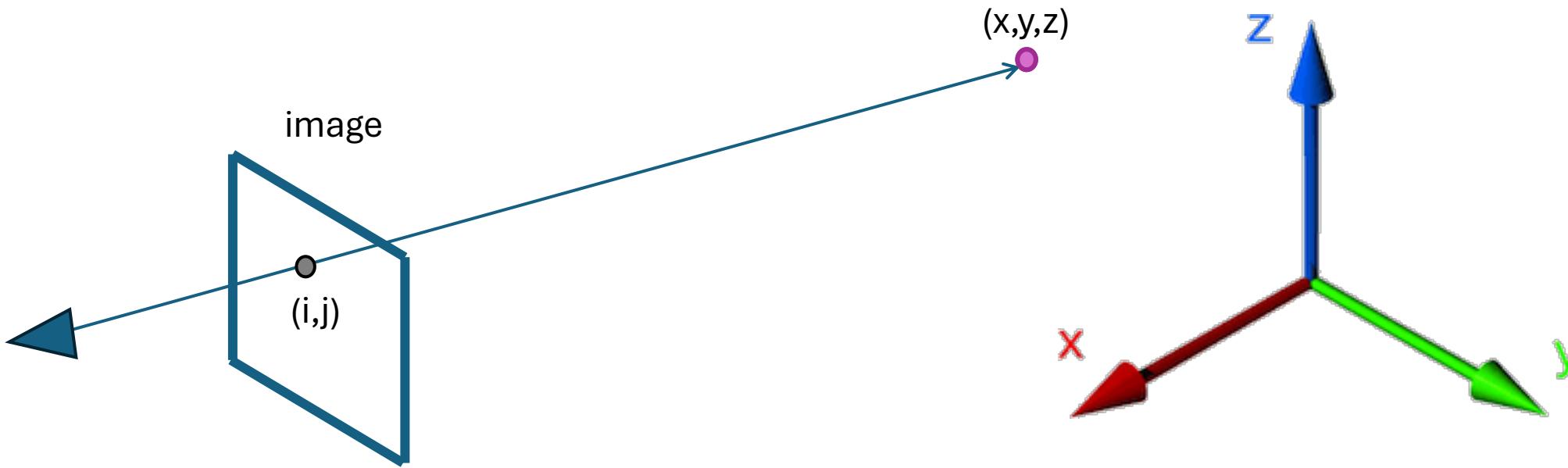
Pointmap: 2D to 3D

# DUST3R: Geometric 3D Vision Made Easy

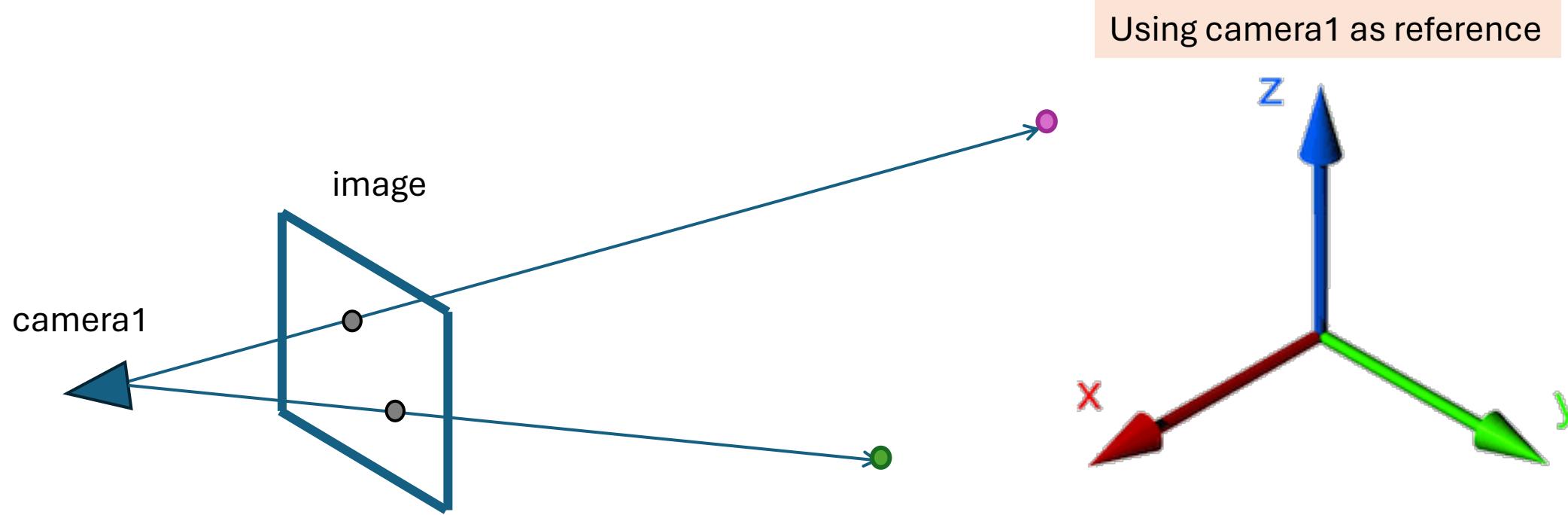


Pointmap: 2D to 3D

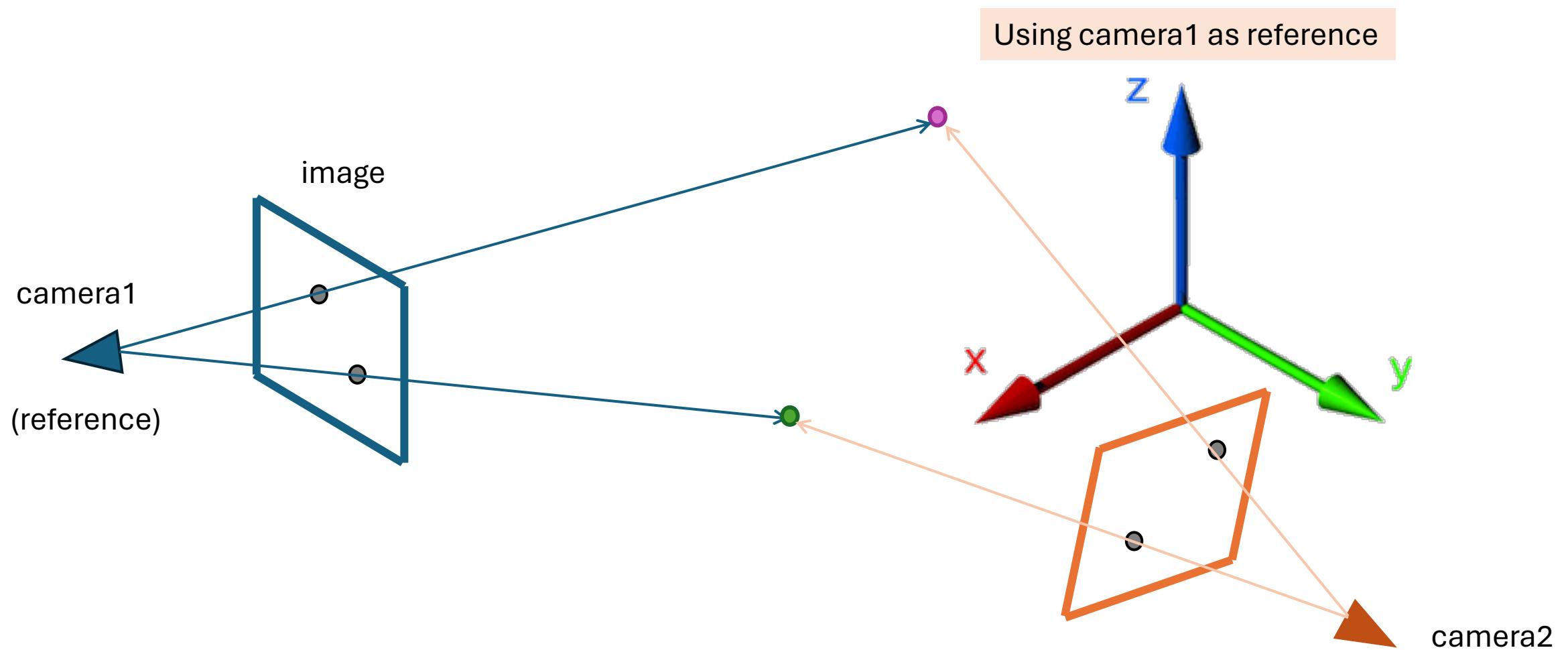
# DUST3R: Geometric 3D Vision Made Easy



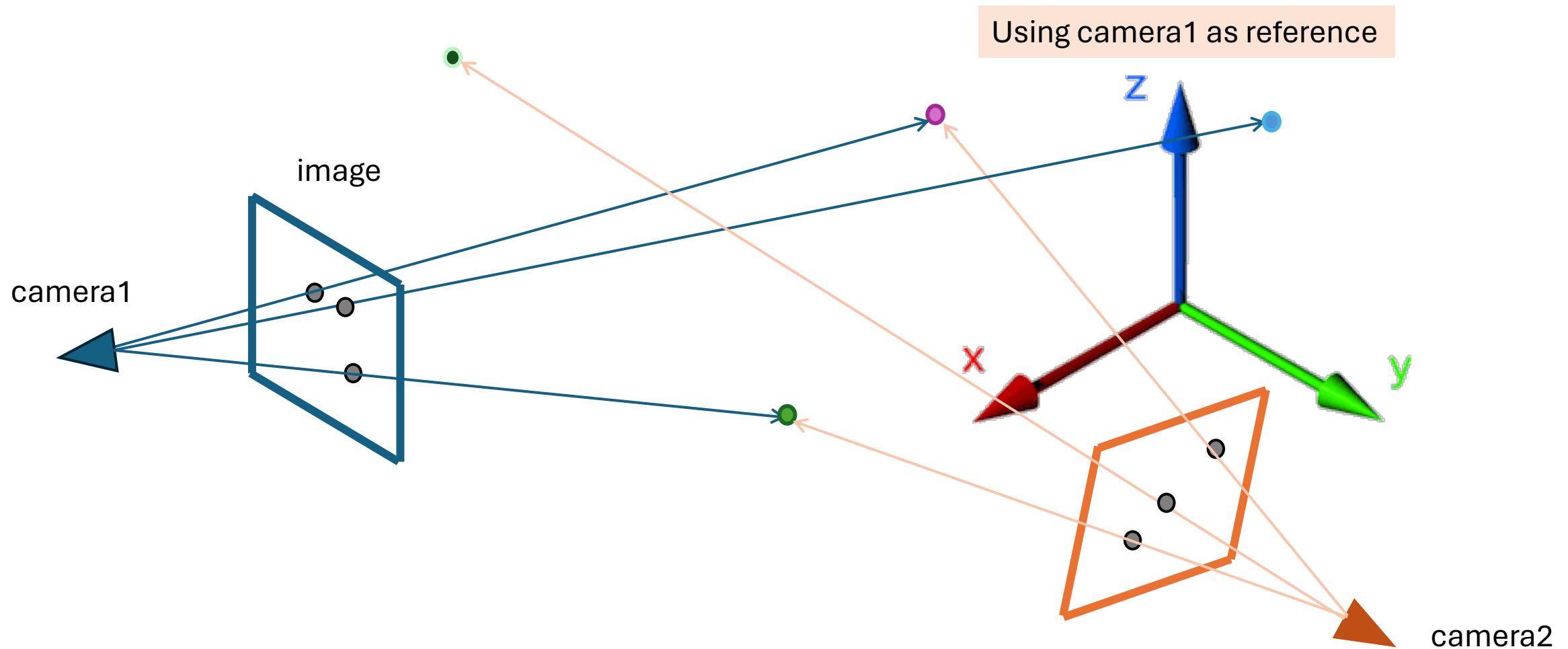
# DUST3R: Geometric 3D Vision Made Easy



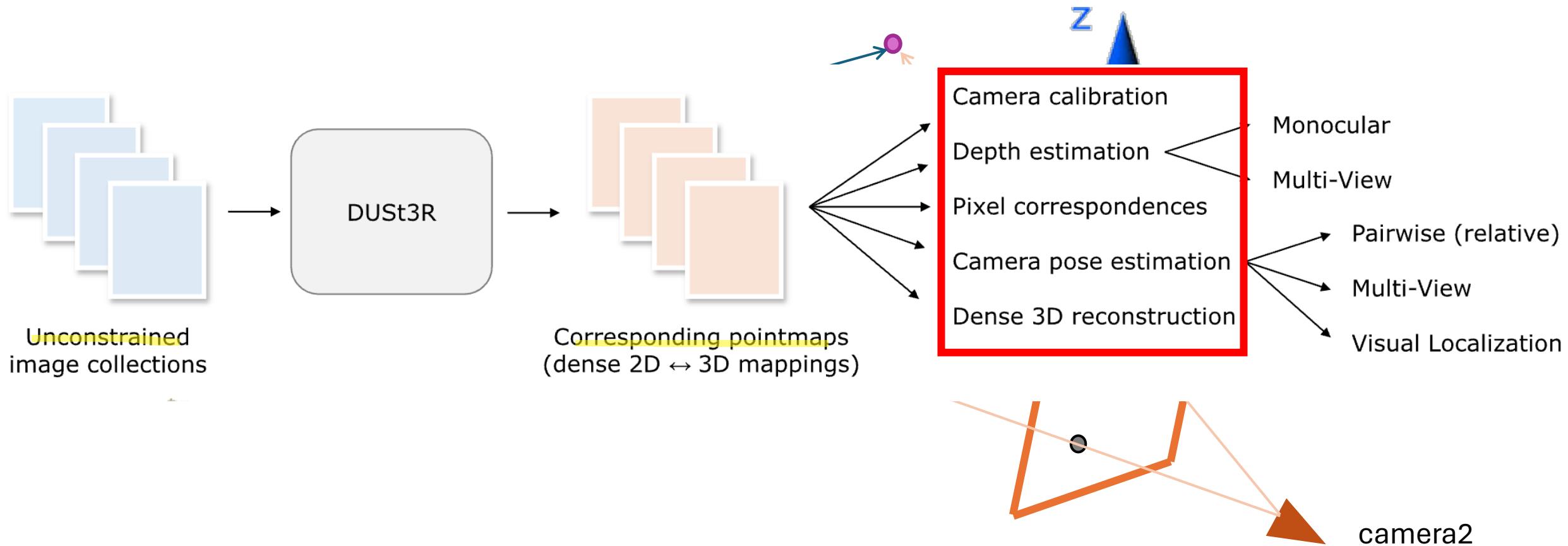
# DUST3R: Geometric 3D Vision Made Easy



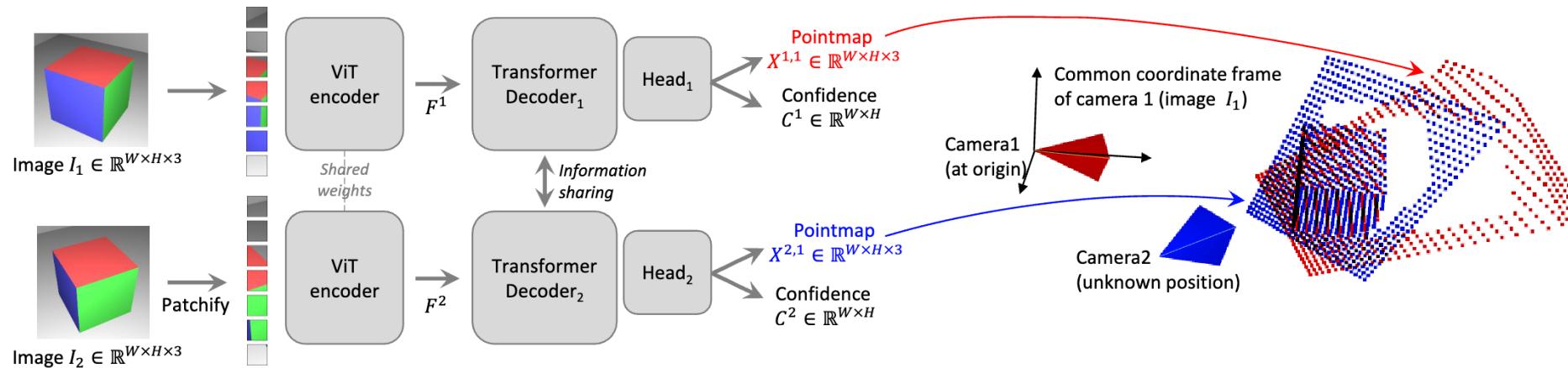
# DUST3R: Geometric 3D Vision Made Easy



# DUST3R: Geometric 3D Vision Made Easy



# DUST3R: Training Objective



The regression loss is defined as the Euclidean distance:

$$\ell_{\text{regr}}(v, i) = \left\| \frac{1}{z} X_i^{v,1} - \frac{1}{\bar{z}} \bar{X}_i^{v,1} \right\|$$

$$v \in \{1,2\}$$

With  $z = \text{norm}(X^{1,1}, X^{2,1})$  and  $\bar{z} = \text{norm}(\bar{X}^{1,1}, \bar{X}^{2,1})$  to handle the scale ambiguity between prediction and ground-truth.

# DUST3R: Applications

## 1. Point Matching

Achieved by mutual nearest neighbour (MNN) search in the **3D space**

$$\mathcal{M}_{1,2} = \{(i, j) \mid i = \text{NN}_1^{1,2}(j) \text{ and } j = \text{NN}_1^{2,1}(i)\}$$

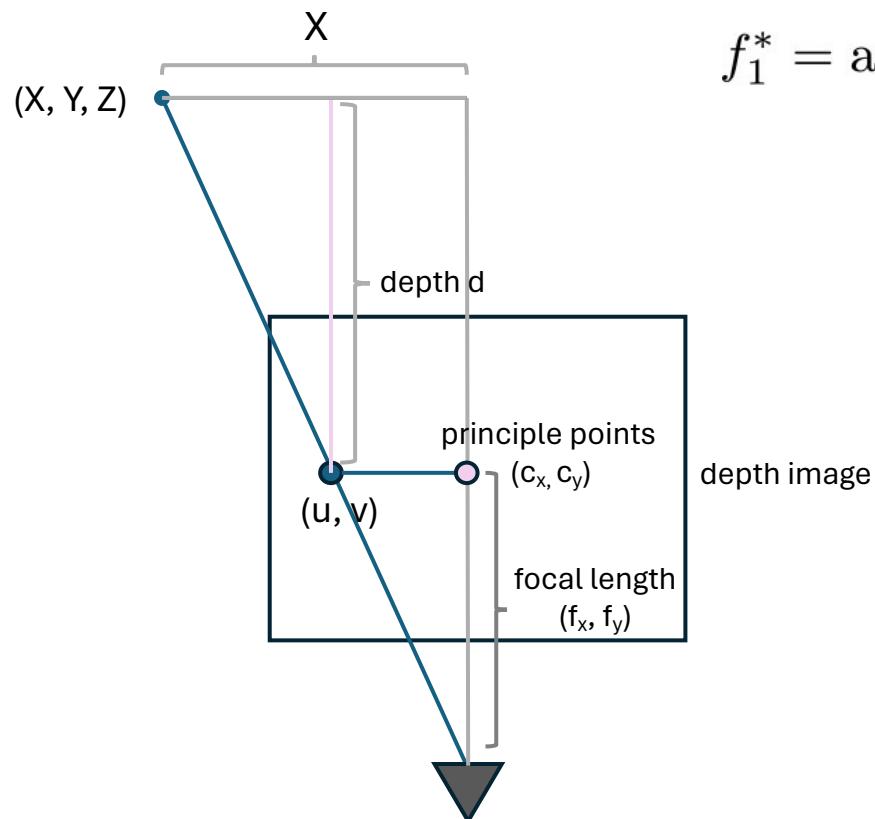
$$\text{with } \text{NN}_k^{n,m}(i) = \arg \min_{j \in \{0, \dots, WH\}} \|X_j^{n,k} - X_i^{m,k}\|.$$



# DUST3R: Applications

## 2. Recovering intrinsic

Estimate the **focal lengths** by **minimize**:



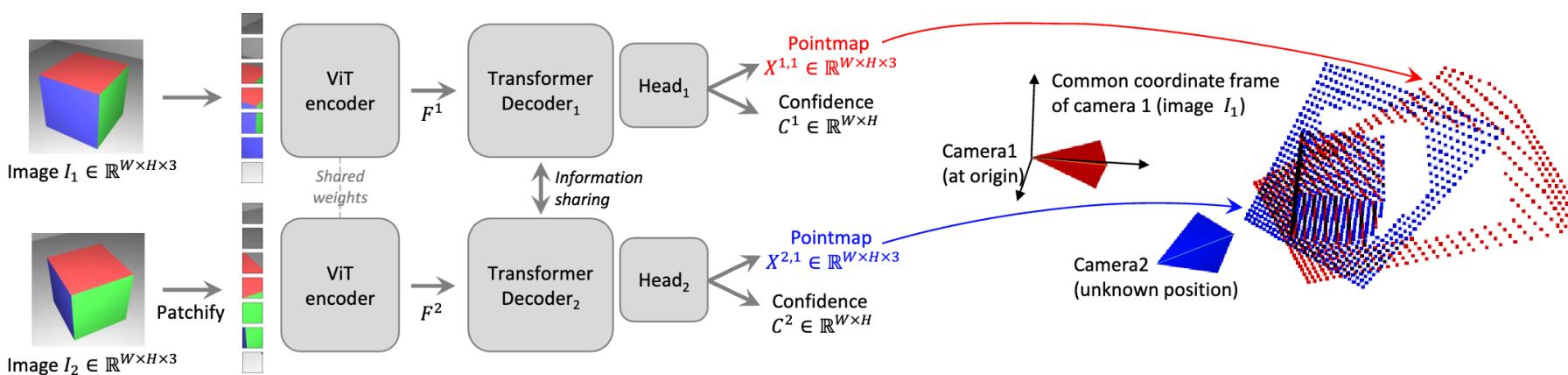
$$f_1^* = \arg \min_{f_1} \sum_{i=0}^W \sum_{j=0}^H C_{i,j}^{1,1} \left\| (i', j') - f_1 \frac{(X_{i,j,0}^{1,1}, X_{i,j,1}^{1,1})}{X_{i,j,2}^{1,1}} \right\|$$

$$\begin{aligned} Z &= \frac{d}{depth\_scale} \\ X &= \frac{(u - c_x) \cdot Z}{f_x} \\ Y &= \frac{(u - c_y) \cdot Z}{f_y} \end{aligned}$$

# DUST3R: Applications

## 3. Relative Camera Pose

### Essential Matrix



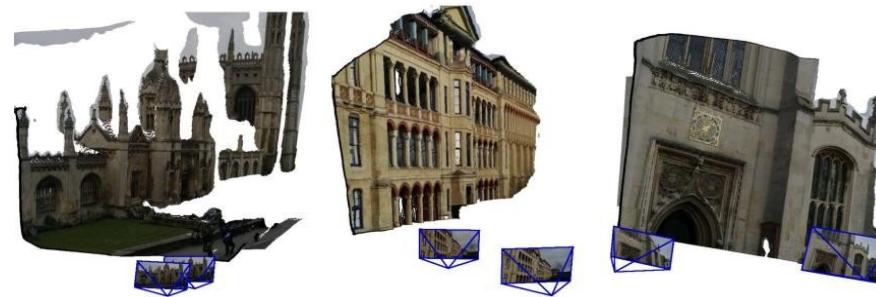
$$x_c'^T E x_c = 0$$

$$E = [t]_{\times} R$$

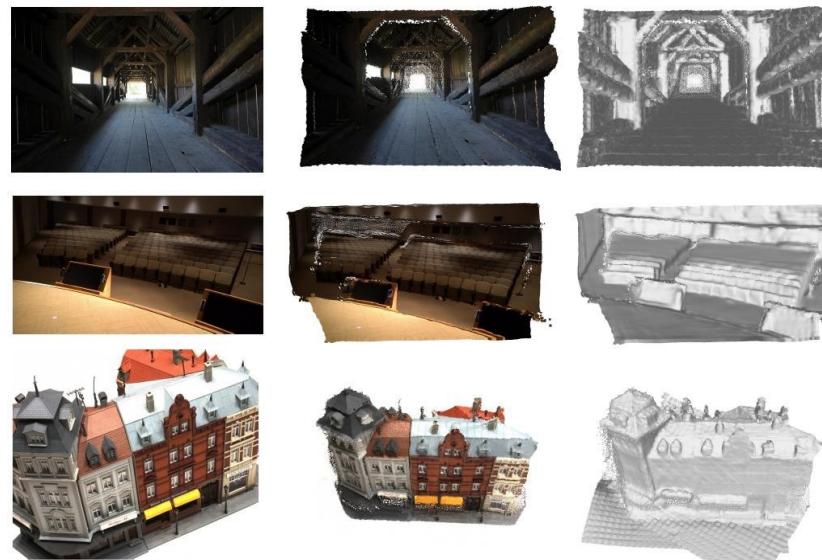
# DUST3R: Applications

## 4. Reconstruction

Two views



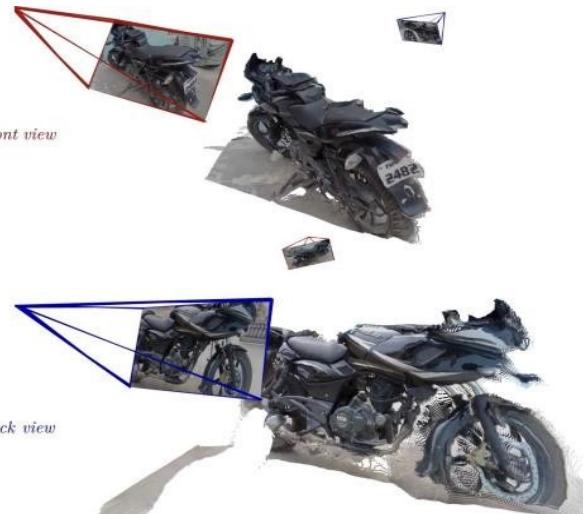
Dense Reconstruction



# DUST3R: Applications

## 4. Reconstruction

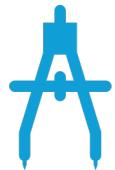
Opposite views



No overlap



# Summary



## Structure From Motion (SfM)

Extending two-view epipolar geometry to multiple views



## Neural Radiance Fields (NeRFs), ECCV 2020

Solves correspondences + triangulation implicitly



## DUSt3R: Geometric 3D Vision Made Easy, CVPR 2024

Deep-learning based method, unify all 3D vision tasks