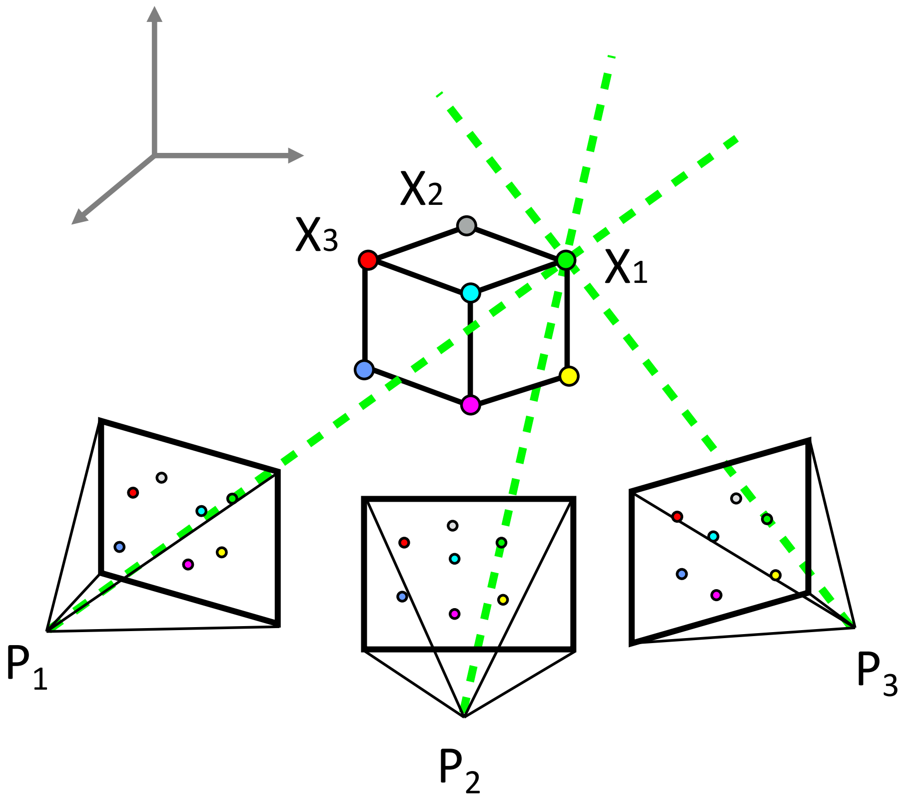
CS 558: Computer Vision 9th Set of Notes

Instructor: Enrique Dunn
Webpage: www.cs.stevens.edu/~edunn
E-mail: edunn@stevens.edu
Office: Lieb 310

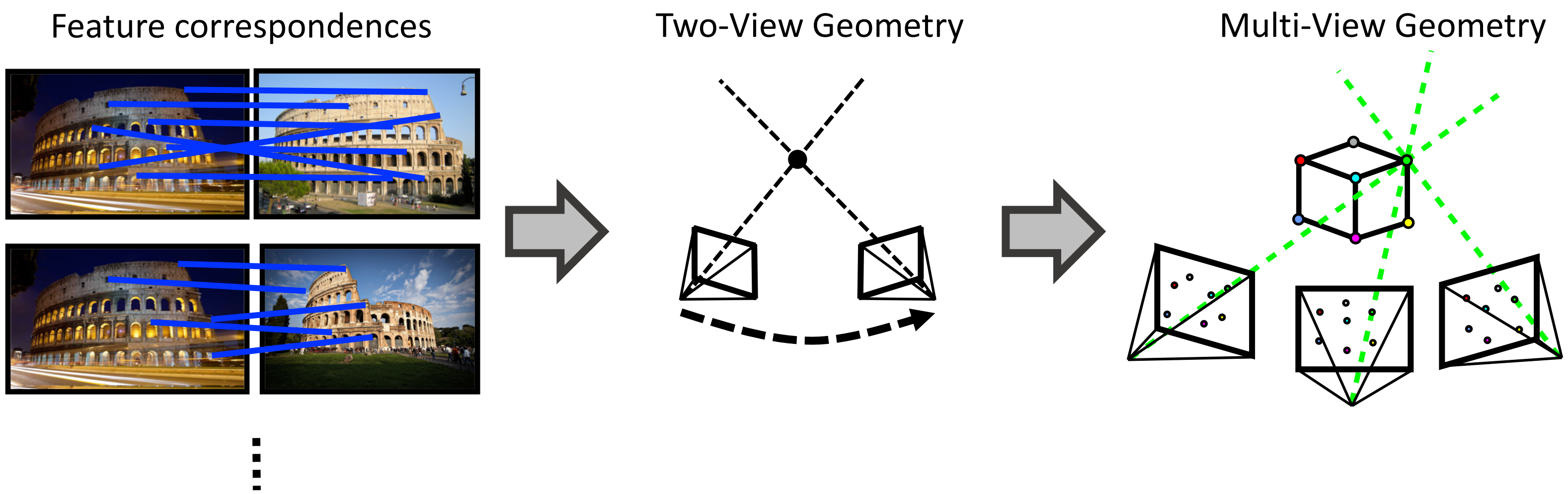
Structure-from-Motion

- Joint estimation of …
  - Structure $\mathbf{X}_i$
  - Cameras $\mathbf{P}_j$
- … from motion, i.e.
  - images at different viewpoints



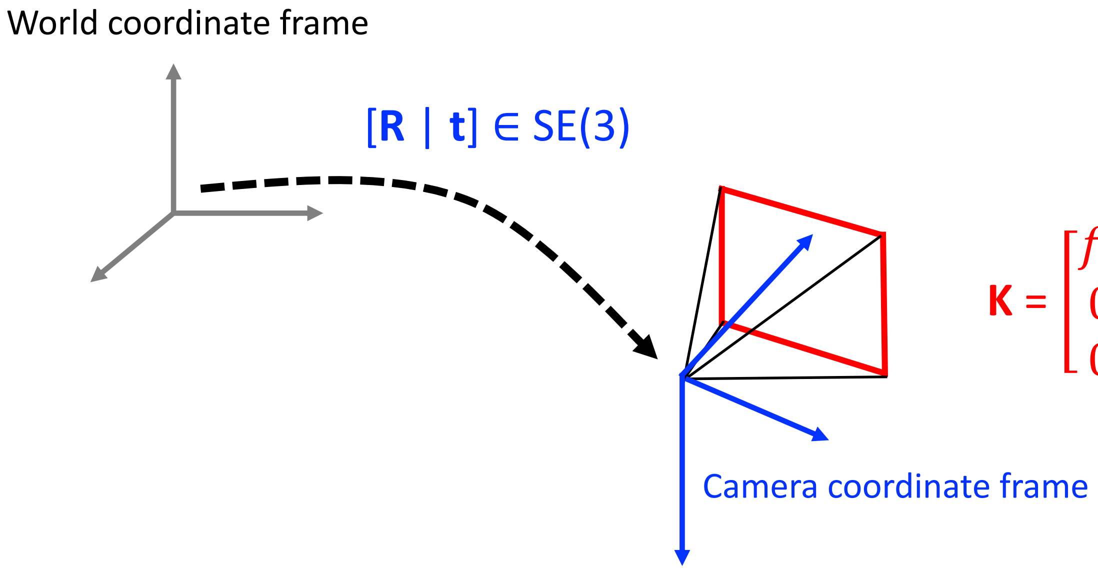
Structure-from-Motion

- Pipeline



Two-View Geometry

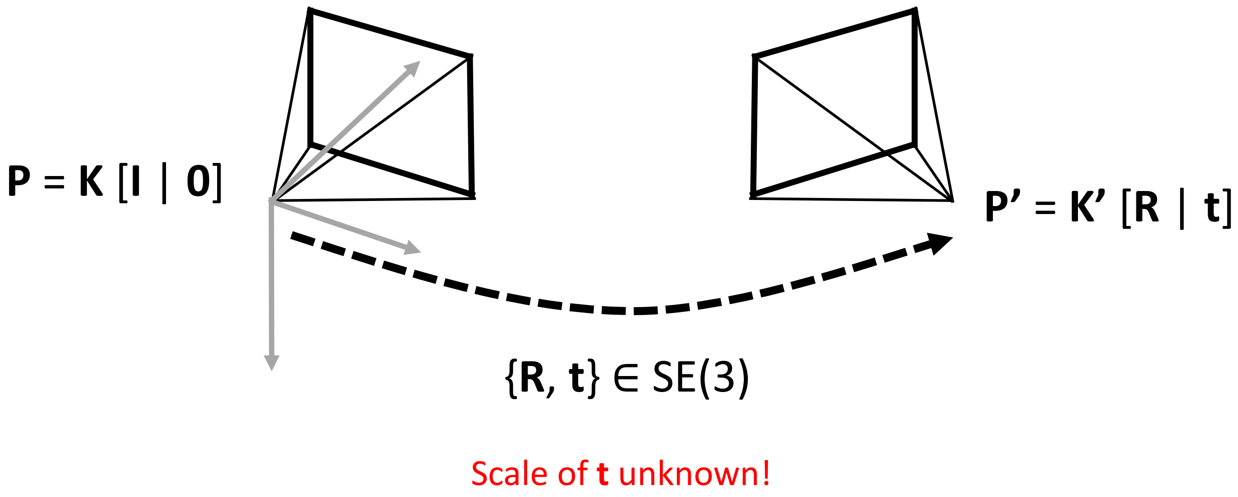
- Absolute camera: **P** = **K** [**R** | **t**]



$$\mathbf{K} = \begin{bmatrix} f_x & \alpha & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

Two-View Geometry

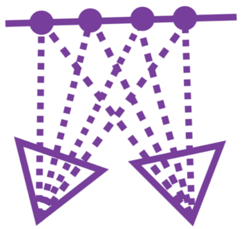
- Relative camera



Two-View Geometry

- Estimation from 2D-2D feature correspondences

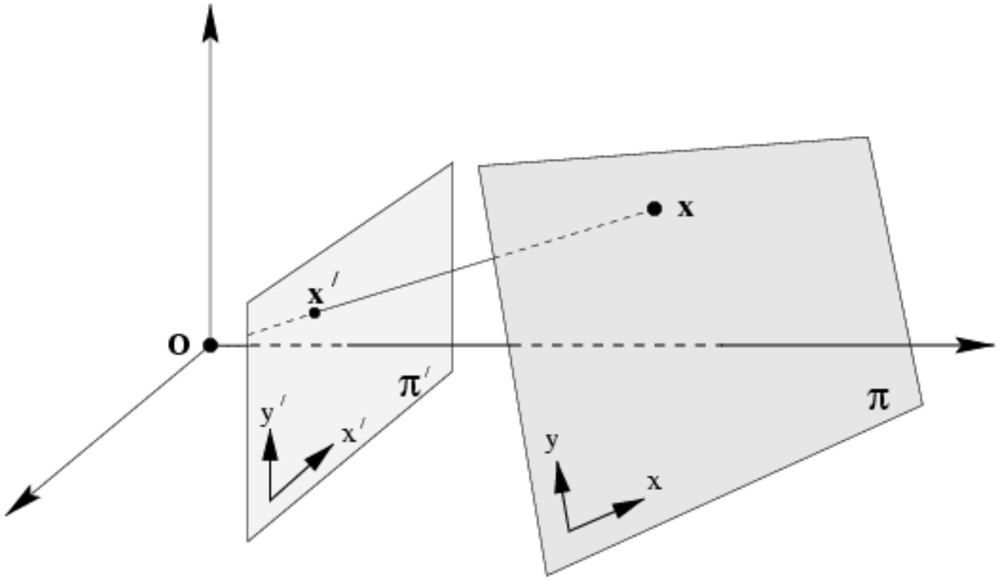
General	Planar	Panoramic
• Fundamental matrix F (*uncalibrated*) • Essential matrix E (*calibrated*)	• Homography H	• Homography H
• 7DOF • 5DOF	• 8DOF	• 8DOF


Hartley and Zisserman 2004, “Multiple View Geometry”

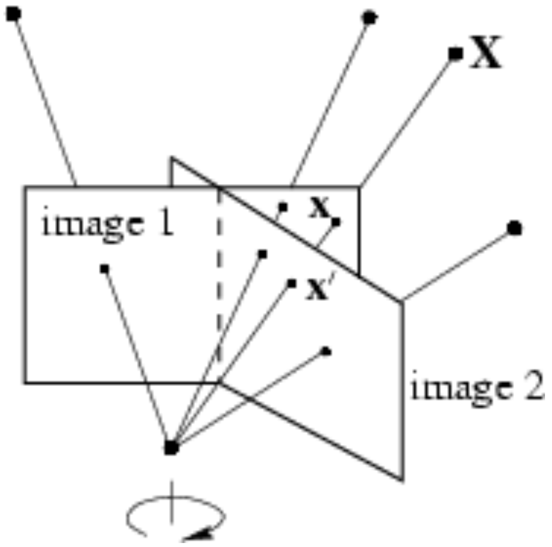
Two-View Geometry

- Homography $\mathbf{H} \in R^{3x3}$
  Linear transformation between two planes in P^2

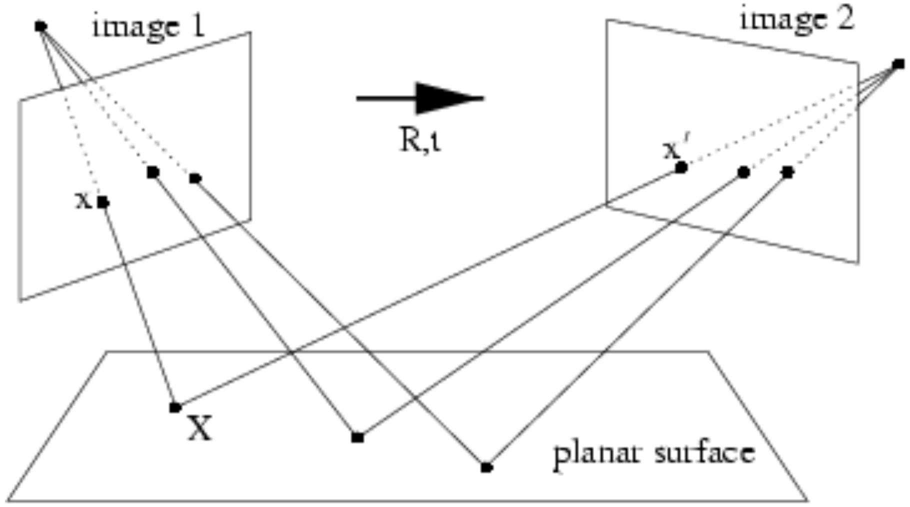


Two-View Geometry

- Homography

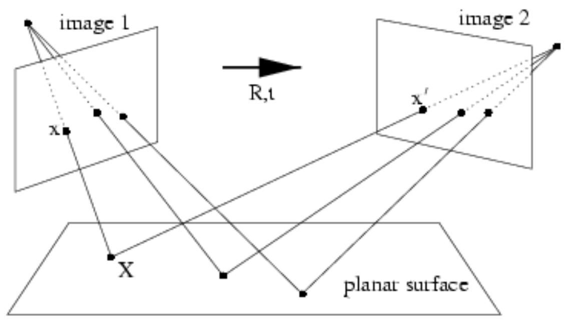


Panoramic



Planar

Two-View Geometry



- Homography
  - In the calibrated case
    - Two cameras: $P = K [I \,|0]$ and $P' = K' [R \mid t]$
    - A plane: $\pi = (n^T, d)^T$
  - The homography is given by $x' = H x$

$$H = K' (R - tn^T/d) K^{-1}$$

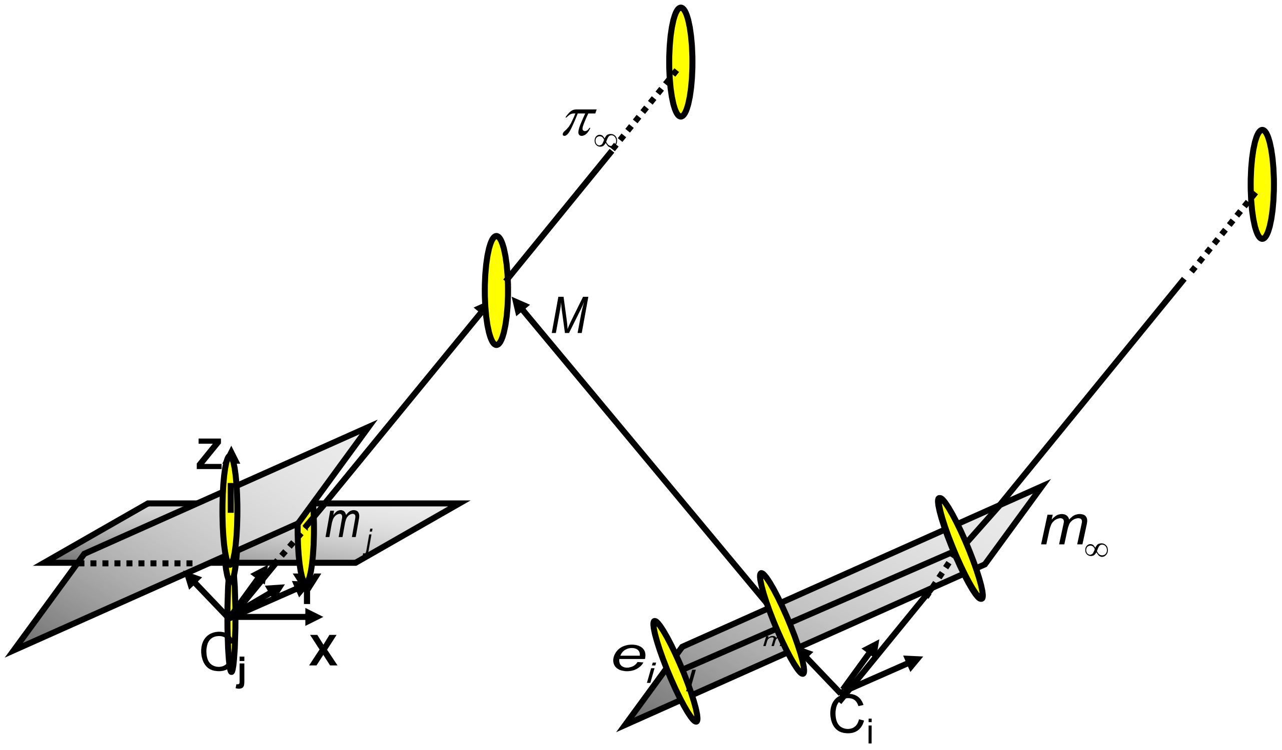
Two-View Geometry

- Fundamental matrix **F**

$$m_\infty = H_\infty m_i$$

$$l = [e]_X m_\infty = \underbrace{[e]_X H_\infty}_{F} m_i$$

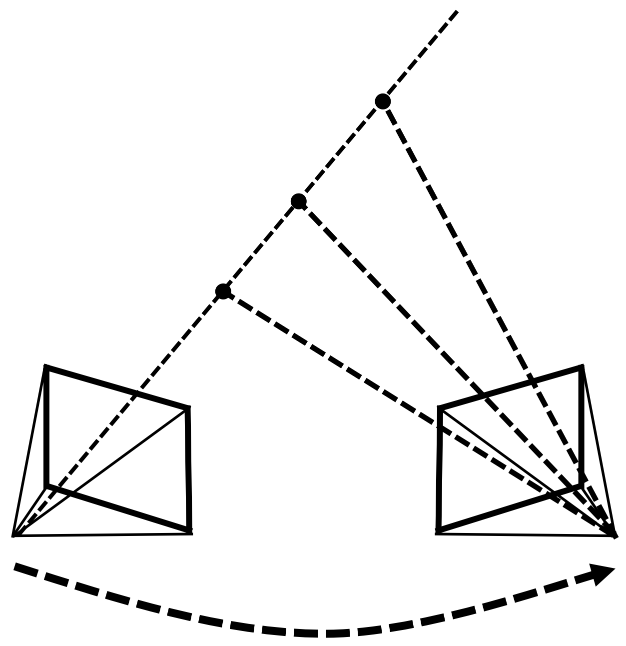
$$m_j l = \underbrace{m_j F m_i = 0}_{\text{epipolar constraint}}$$



Two-View Geometry

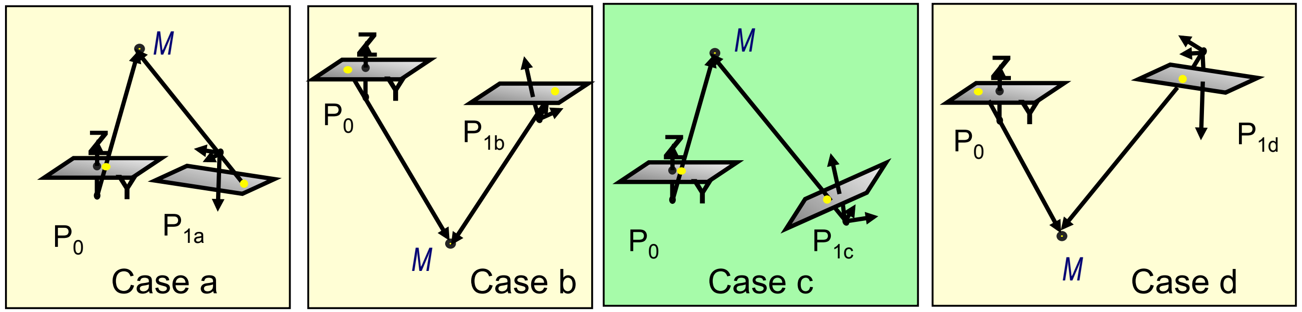
- Essential matrix **E**

$$\mathbf{E} = \mathbf{K}^{\mathrm{T}}\,\mathbf{F}\,\mathbf{K'} = \mathbf{t}_{\mathrm{x}}\,\mathbf{R}$$



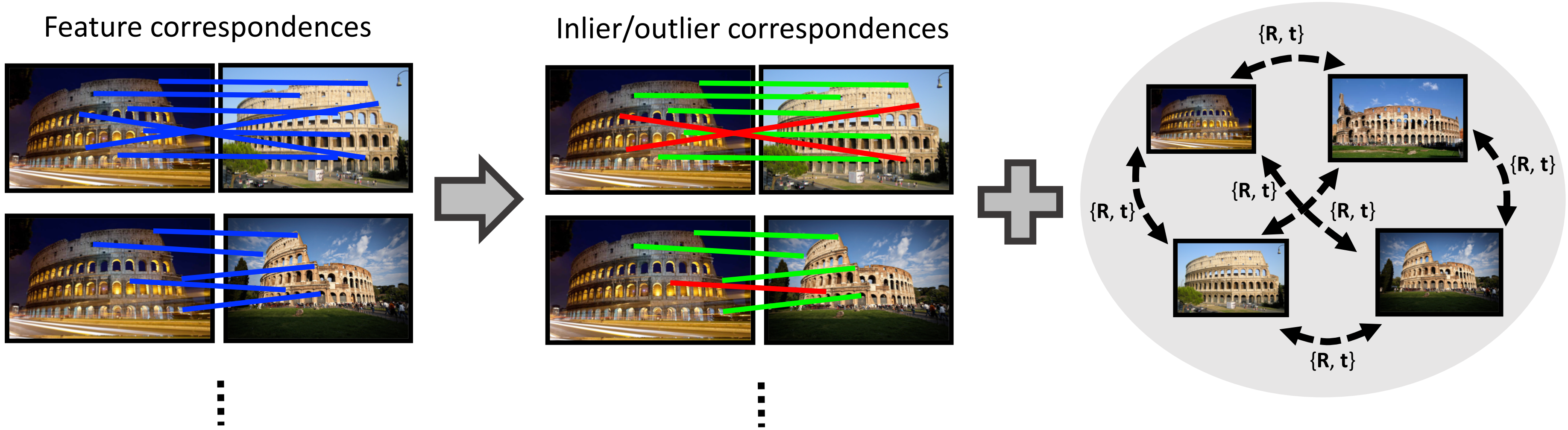
Two-View Geometry

- Robust estimation using RANSAC
- Model selection (**F**, **E**, **H**)
  - Inlier maximization
  - GRIC criterion Torr 1998, “An assessment of information criteria for motion model selection"
  - QDEGSAC Frahm and Pollefeys 2006, “RANSAC for (quasi-) degenerate data (QDEGSAC)”
- Decompose **E**, **H** to {**R**, **t**}



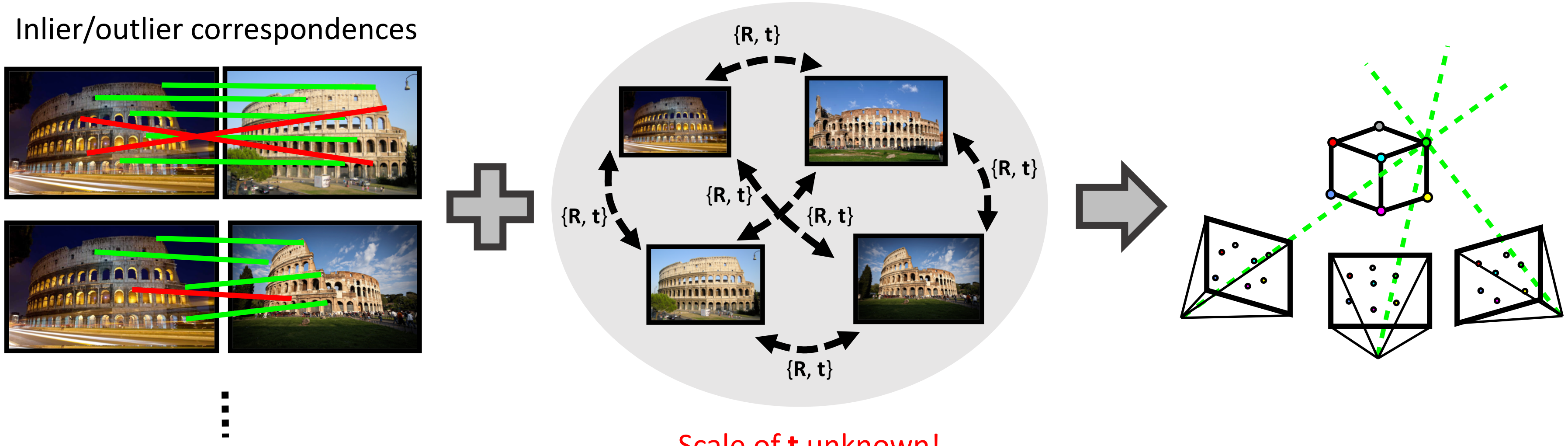
Two-View Geometry

- “Augmented” scene graph



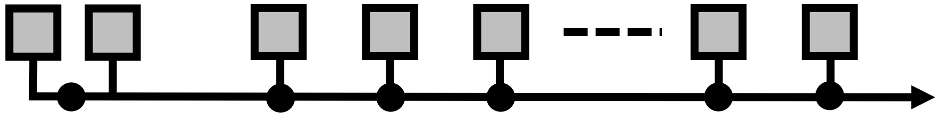
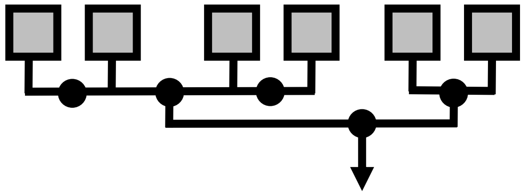
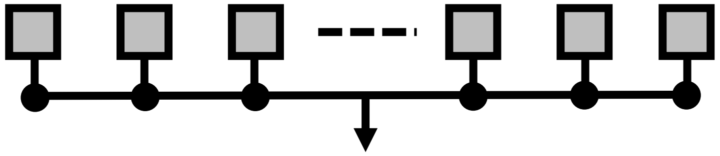
Structure-from-Motion

- From relative to absolute cameras and structure



Scale of **t** unknown!

Structure-from-Motion

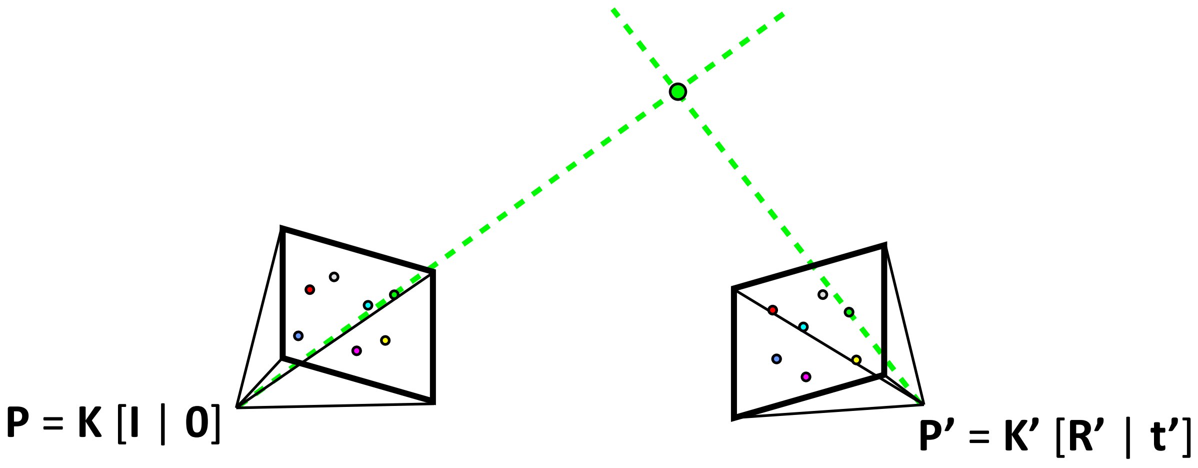
- 3 paradigms
  - Incremental
    
  - Hierarchical
    
  - Global
    

Incremental SfM

- Initialization
  1. Choose two non-panoramic views ($\|t\| = 1$)

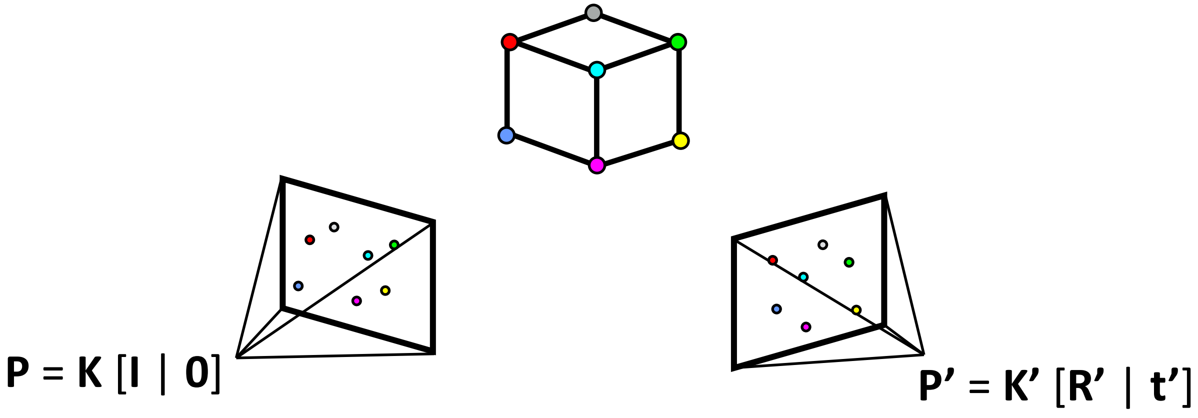
Incremental SfM

- Initialization
  1. Choose two non-panoramic views ($\|t\| = 1$)
  2. Triangulate inlier correspondences



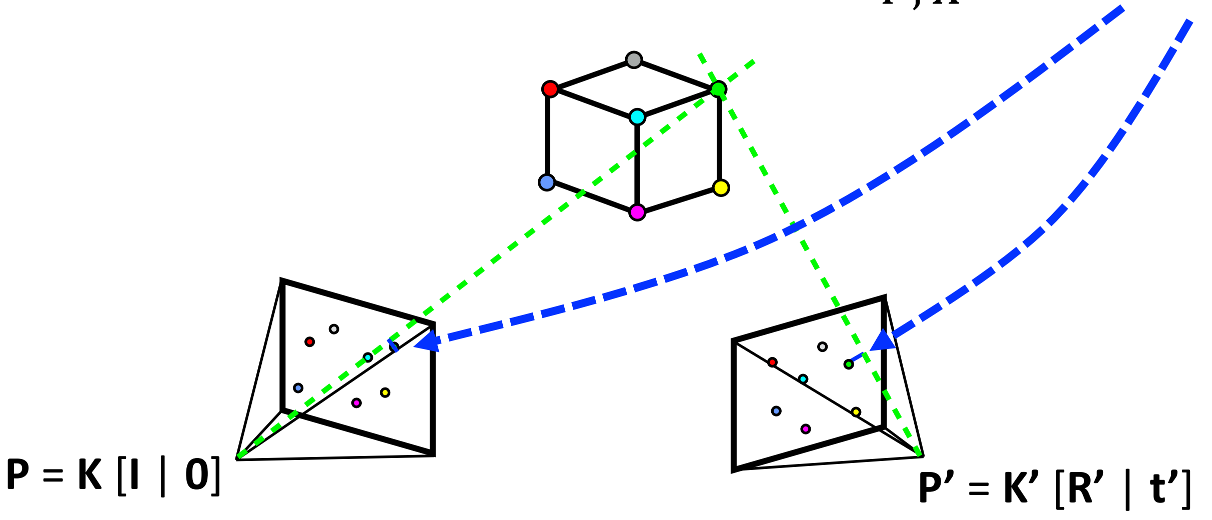
Incremental SfM

- Initialization
  1. Choose two non-panoramic views ($\|t\| = 1$)
  2. Triangulate inlier correspondences
  3. Bundle adjustment



Incremental SfM

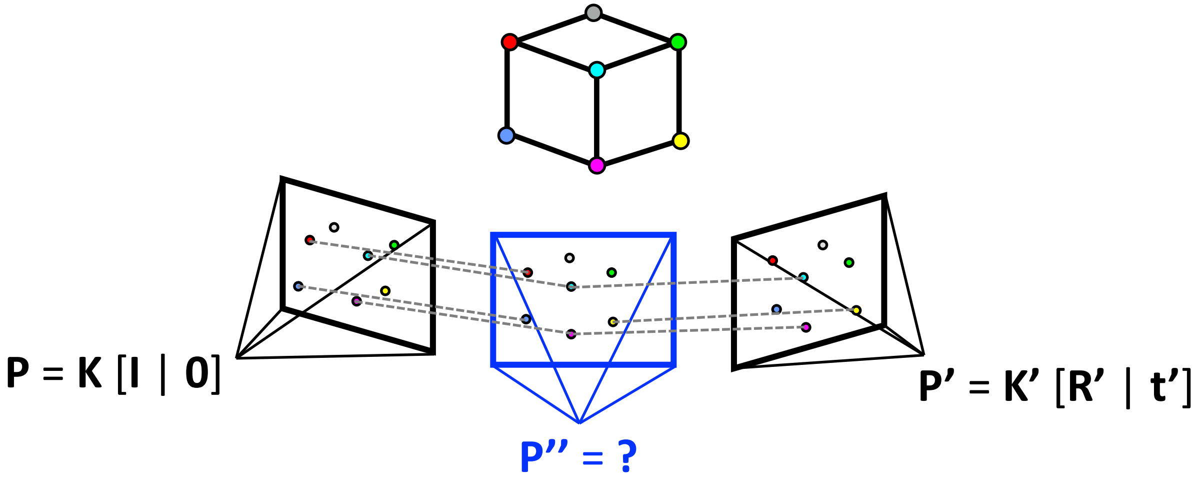
- Bundle adjustment
  - Non-linear refinement of structure and motion
  - Minimize reprojection error: $\min_{P,X} \|x - \pi(P, X)\|$



Triggs et al. 1999, "Bundle Adjustment – A Modern Synthesis"

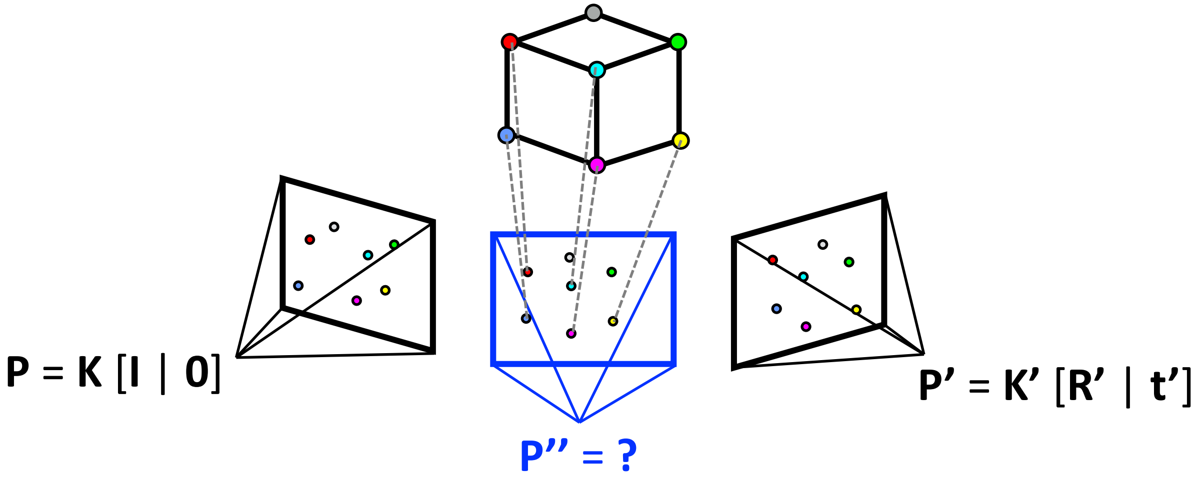
Incremental SfM

- Absolute camera registration
    1. Find 2D-3D correspondences



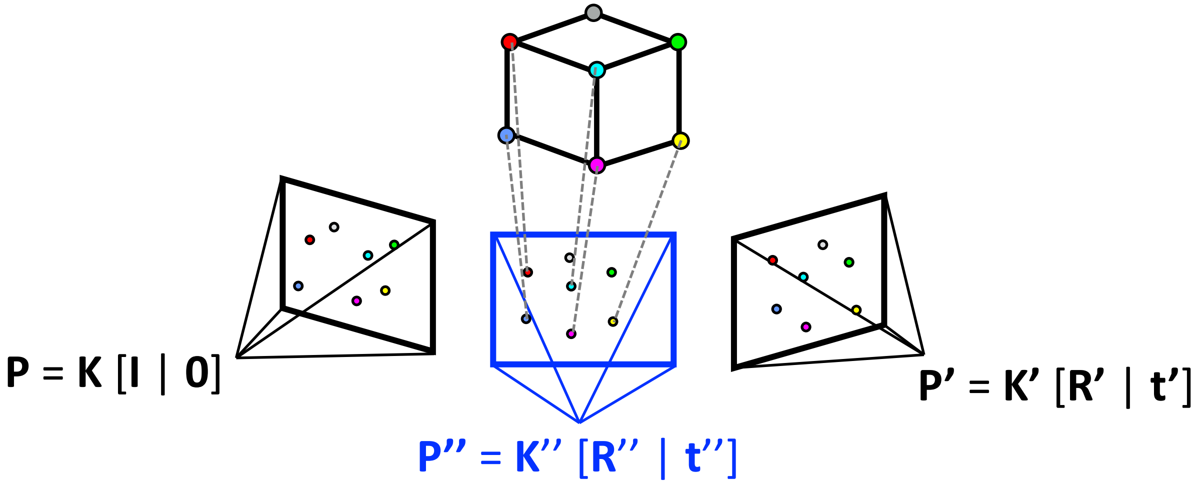
Incremental SfM

- Absolute camera registration
    1. Find 2D-3D correspondences

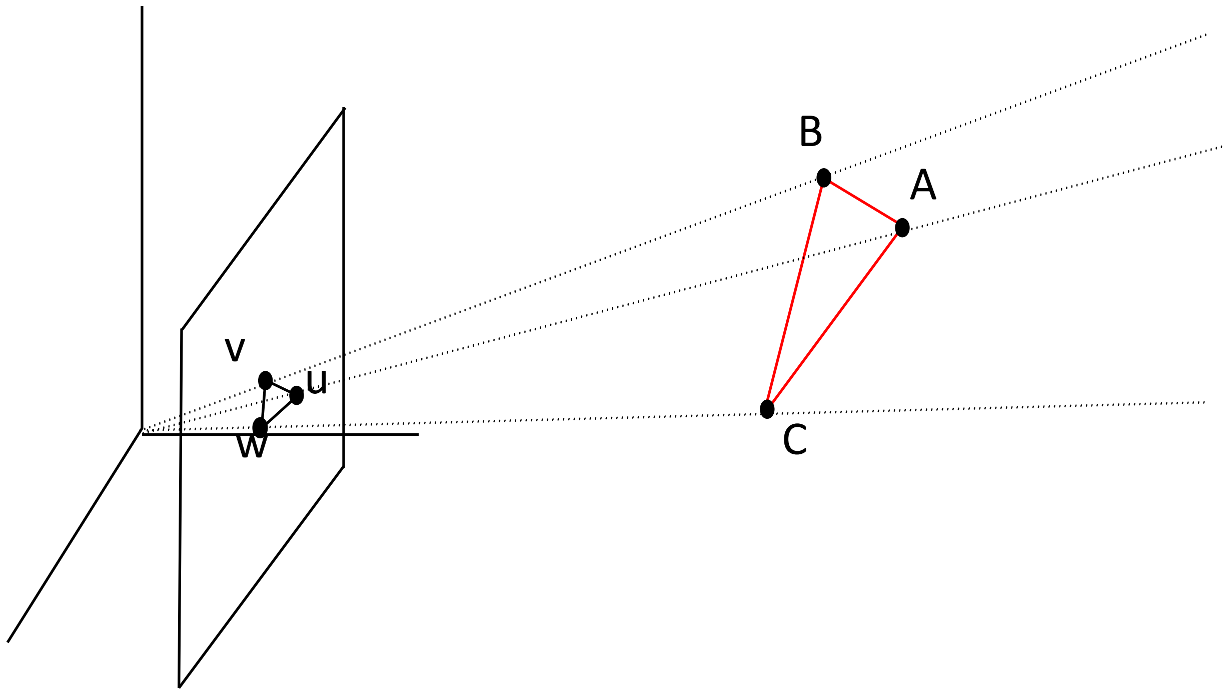


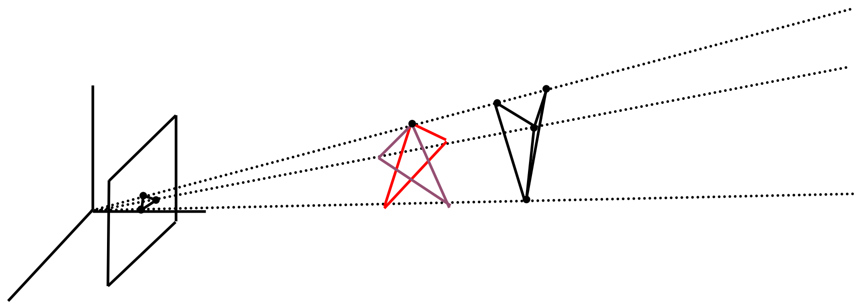
Incremental SfM

- Absolute camera registration
  1. Find 2D-3D correspondences
  2. Solve Perspective-n-Point problem



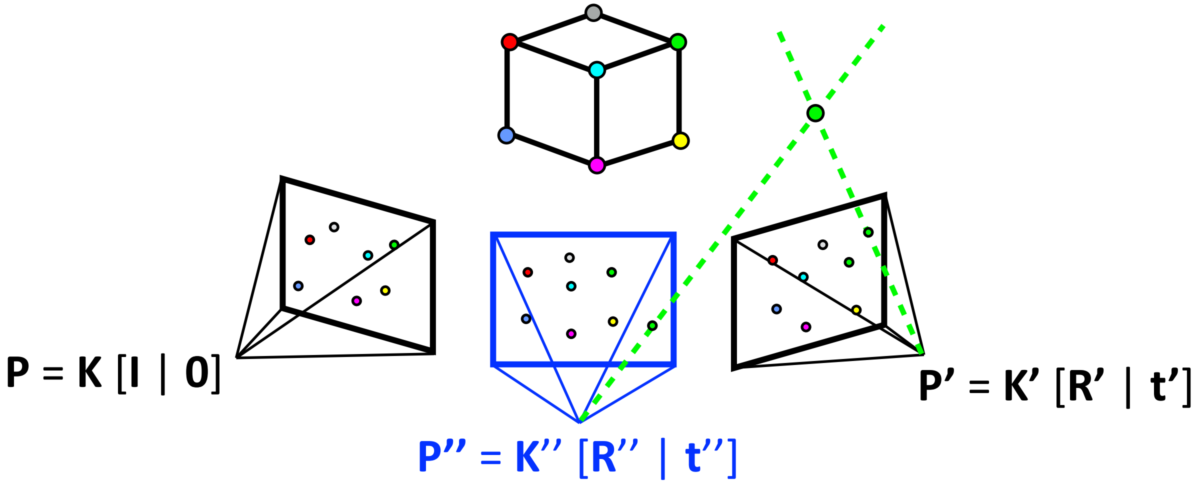
Perspective-3-Point Problem





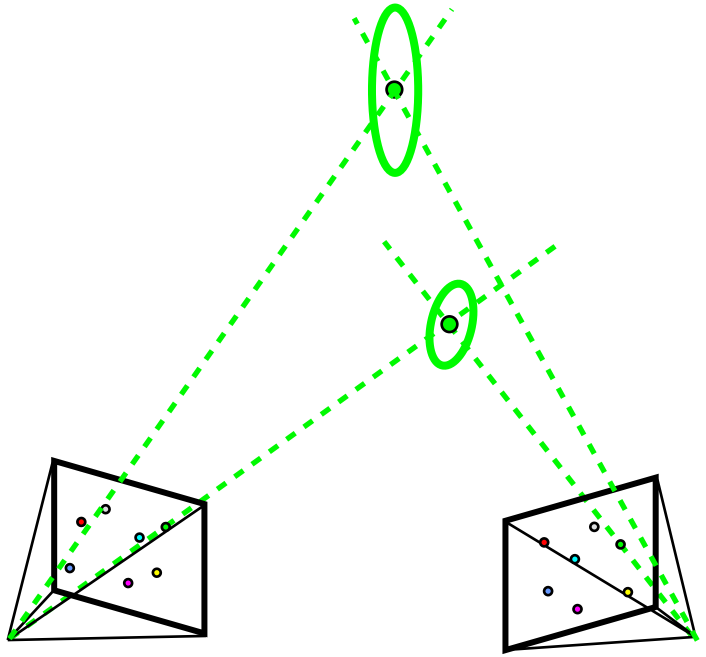
Incremental SfM

- Absolute camera registration
  1. Find 2D-3D correspondences
  2. Solve Perspective-n-Point problem
  3. Triangulate new points

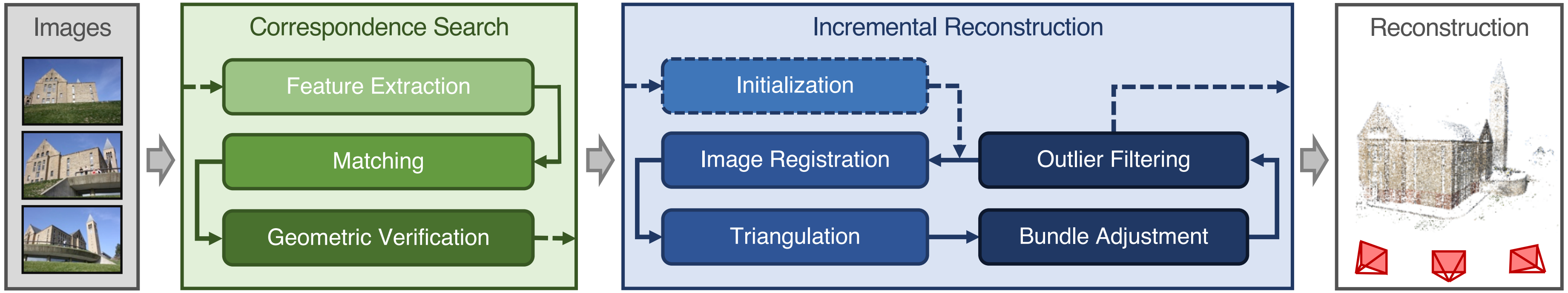


Incremental SfM

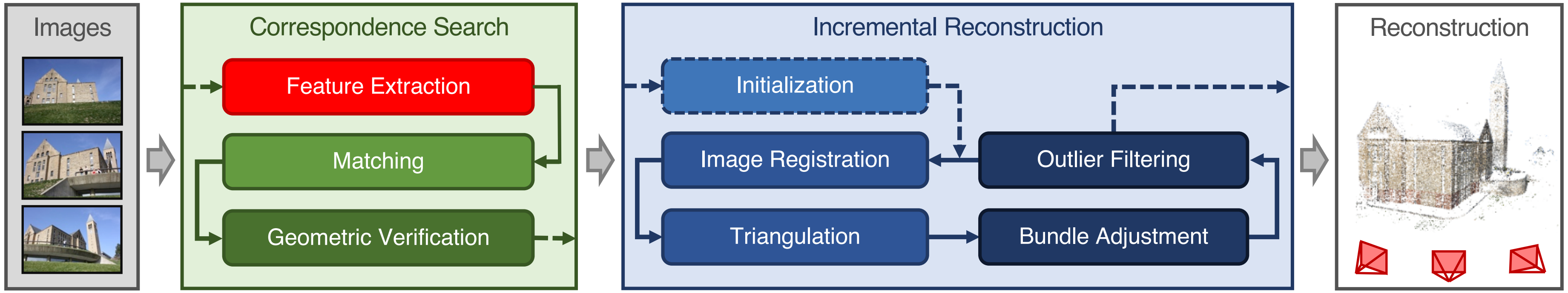
- Outlier filtering
  - Remove points with large reprojection error
  - Remove points at “infinity”



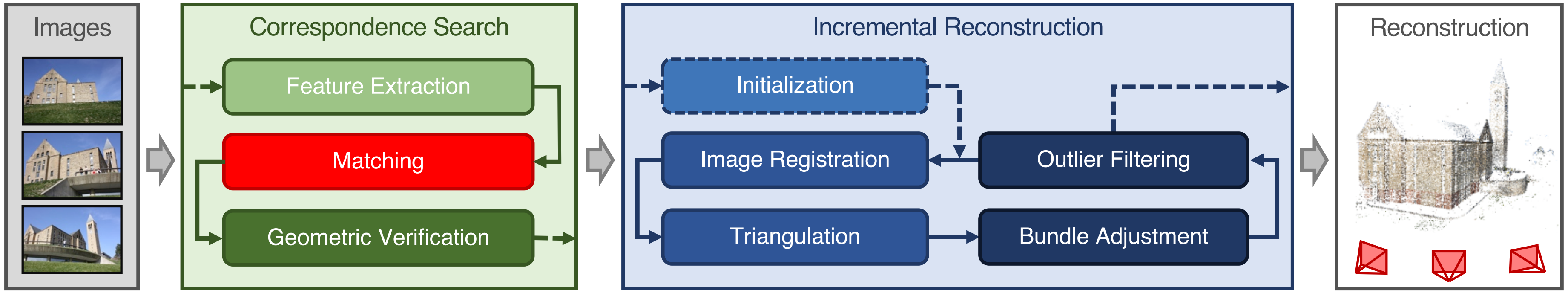
Incremental SfM

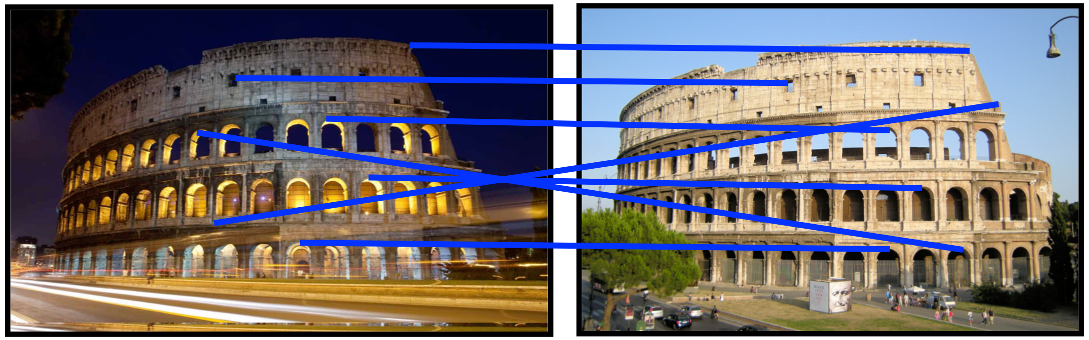


Incremental SfM

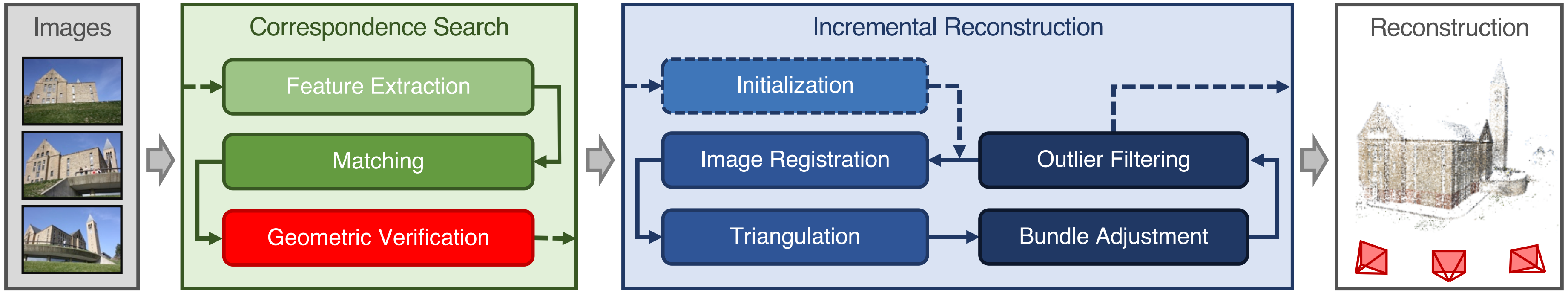


Incremental SfM

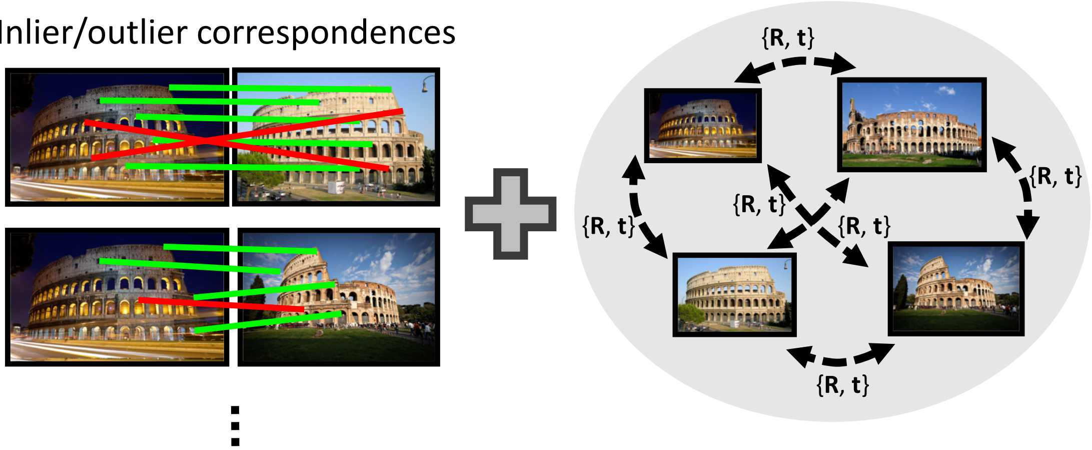




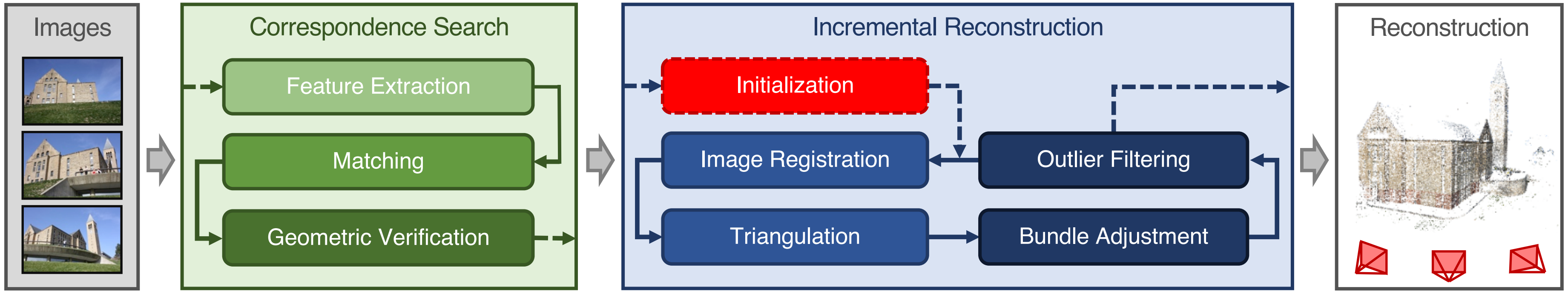
Incremental SfM

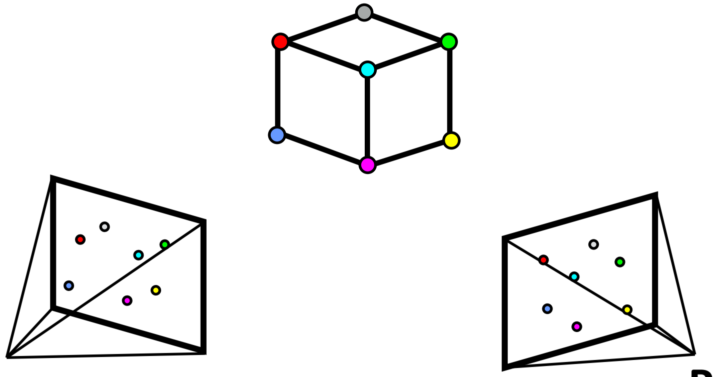


Inlier/outlier correspondences



Incremental SfM

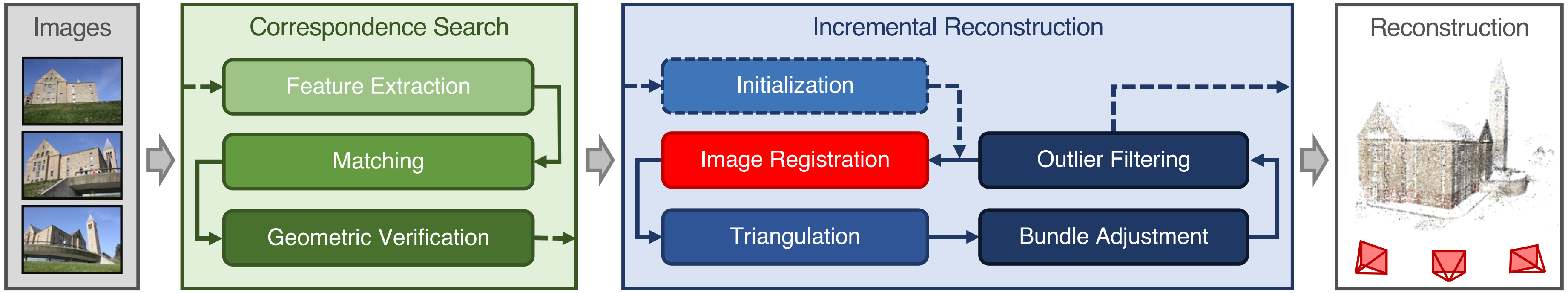


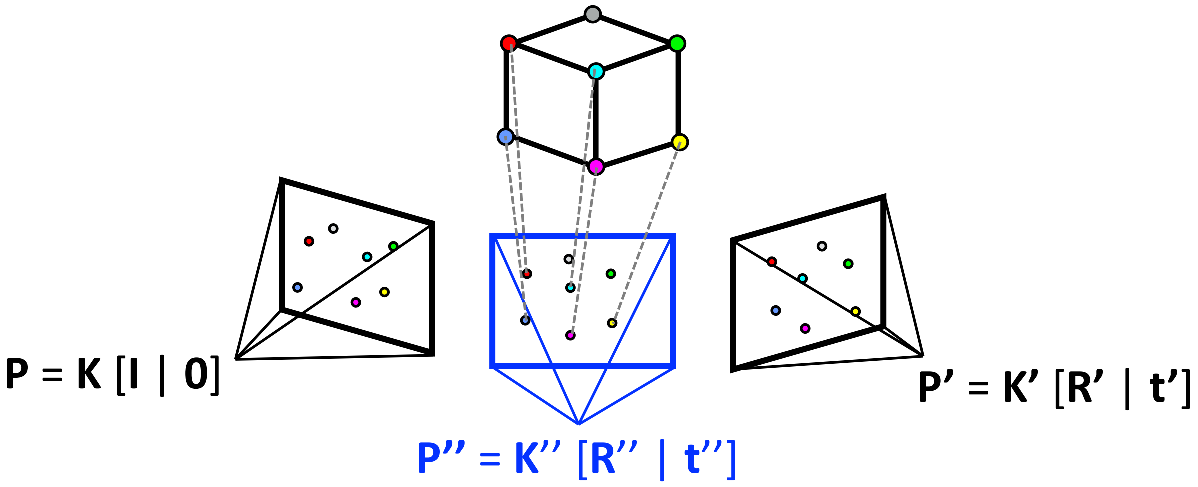


$P = K\,[I \mid 0]$

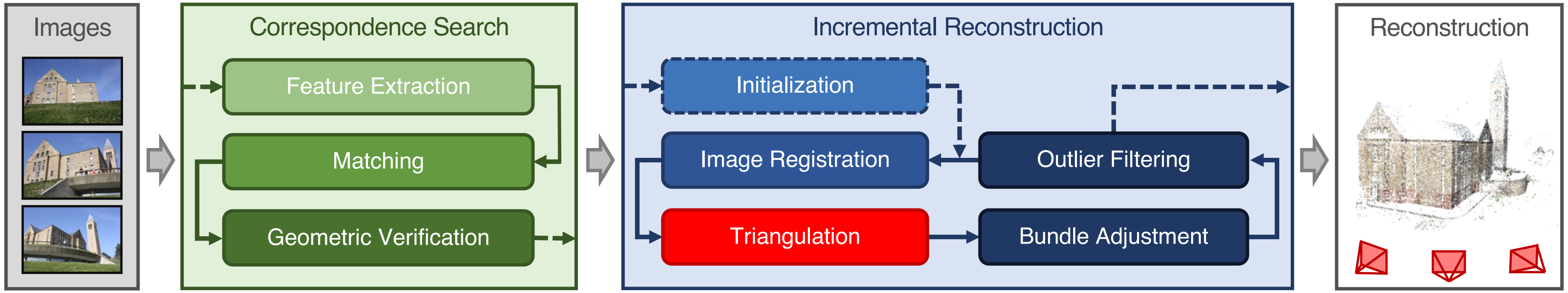
$P' = K'\,[R' \mid t']$

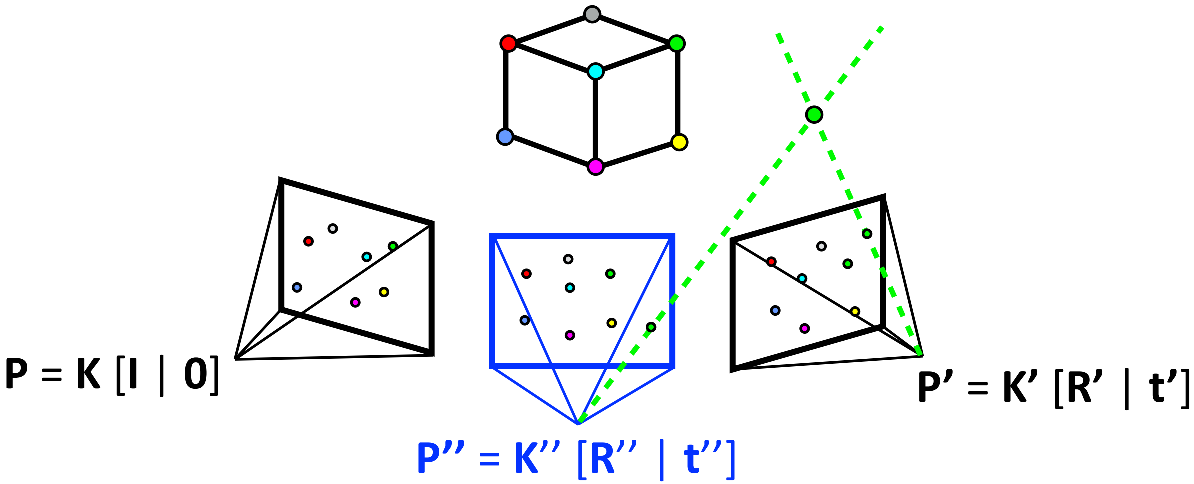
Incremental SfM



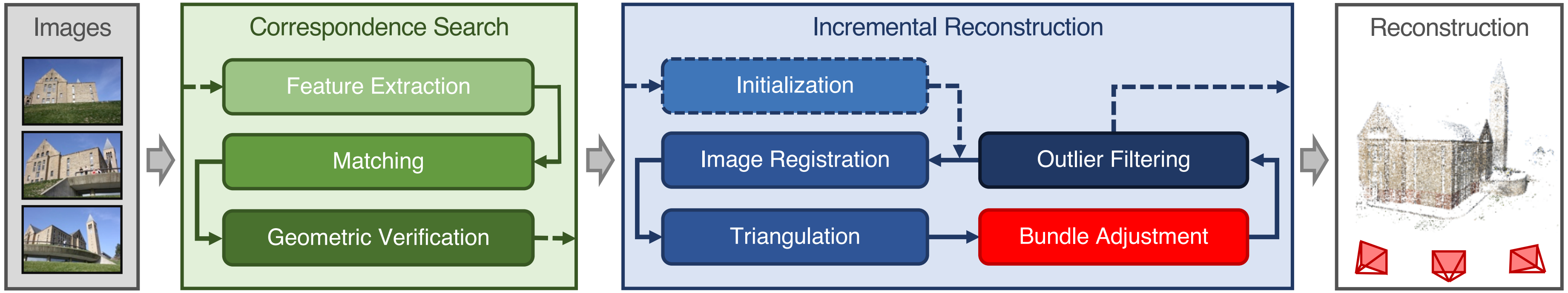


Incremental SfM



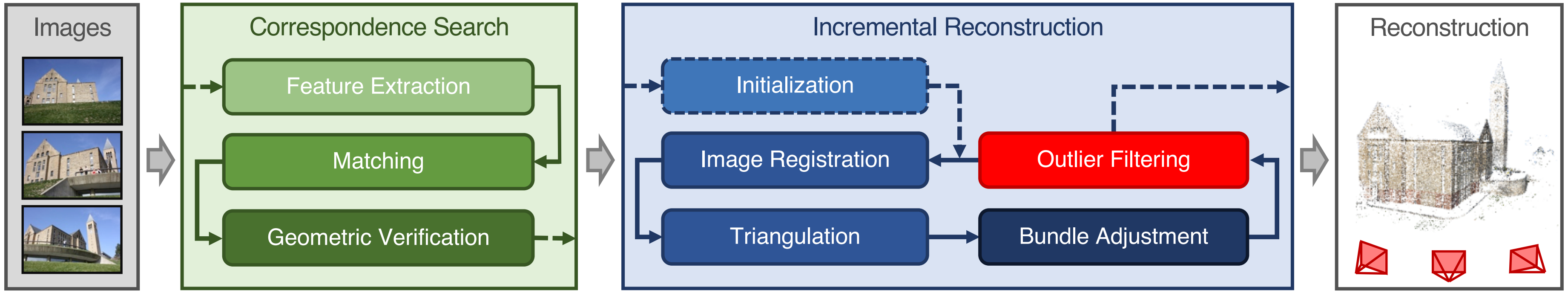


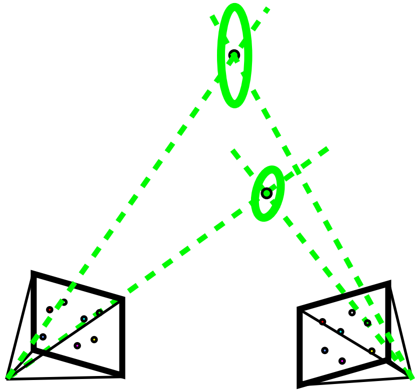
Incremental SfM



$$\min_{P,X} \|x - \pi(P, X)\|$$

Incremental SfM





Overview

- Basic recognition tasks
- A machine learning approach
  - Example features
  - Example classifiers
  - Levels of supervision
  - Datasets
- Current trends and advanced recognition tasks

Specific recognition tasks

Image parsing

Scene Categorization:
-outdoor/indoor
-city/forest/factory/etc.

Image annotation/tagging
-sky, people, mountain, etc.

Object detection
-find pedestrians

Activity recognition
-walking
-shopping
-sitting
-rolling a cart

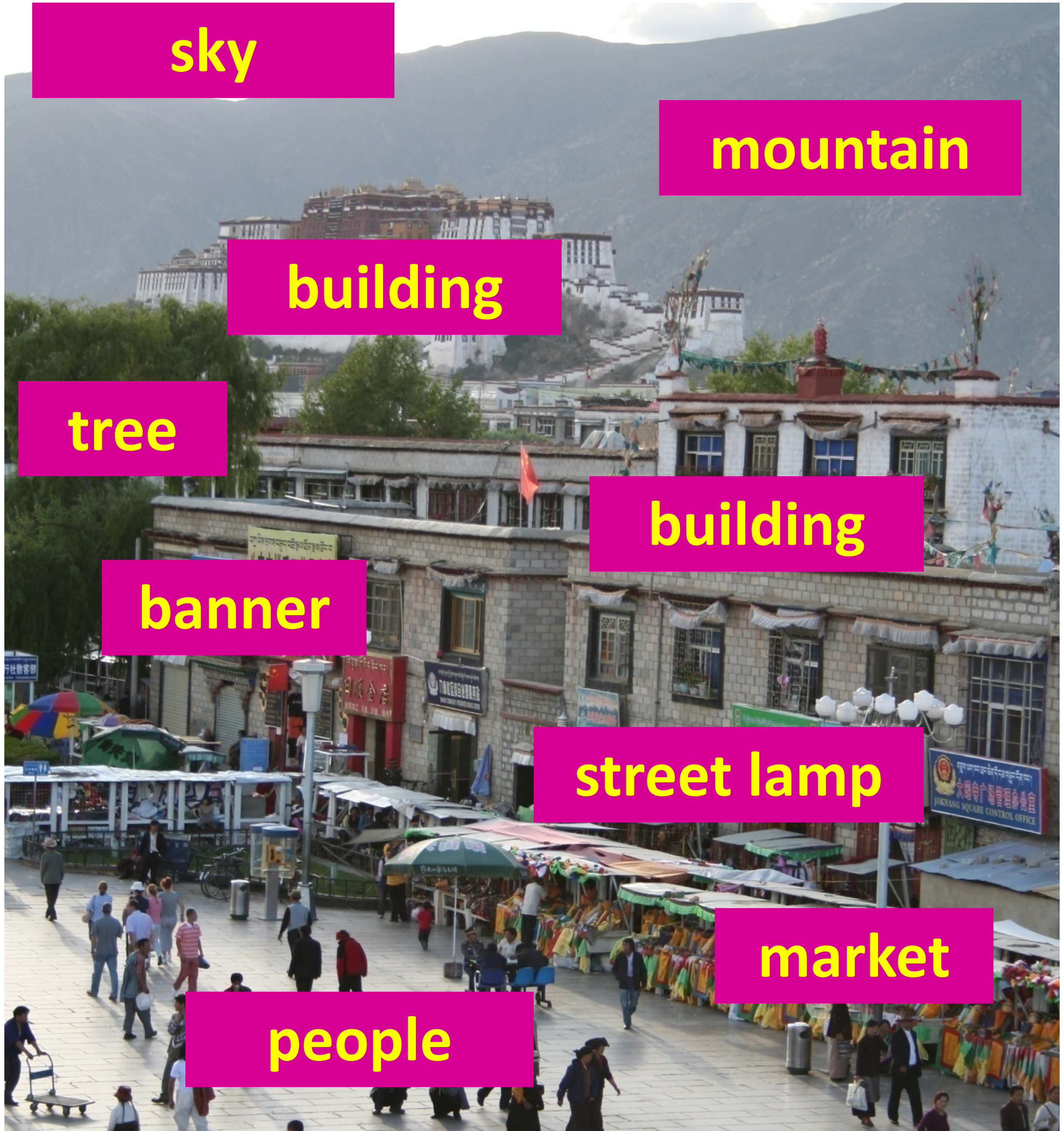


Image understanding?

The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

The machine learning framework

$$\mathbf{y} = \mathbf{f}(\mathbf{x})$$

output prediction function Image feature

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Steps

Training

Training Images

Image Features → Training → Learned model

Training Labels → Training

Testing

Test Image

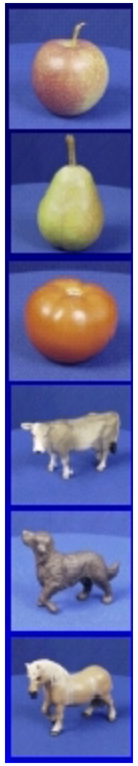
Image Features → Prediction

Learned model → Prediction

Slide credit: D. Hoiem

Generalization

Training set (labels known)



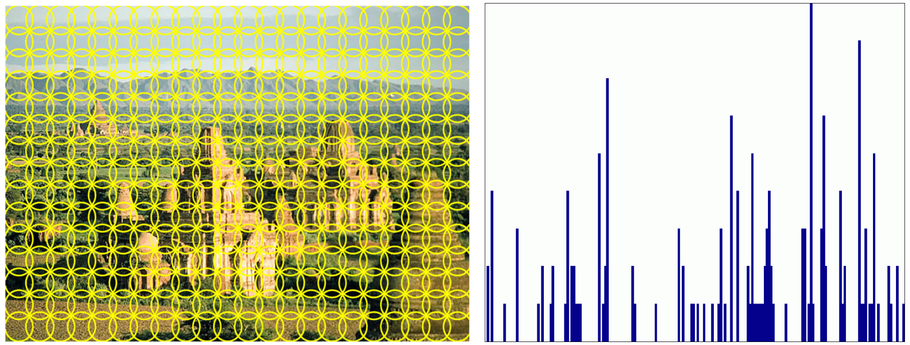
Test set (labels unknown)

- How well does a learned model *generalize* from the data it was trained on to a new test set?

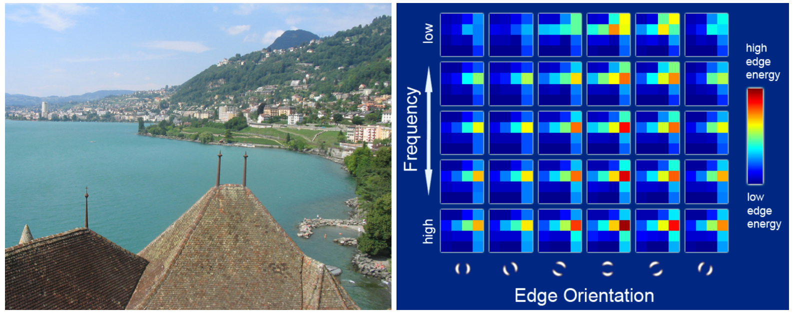
Popular global image features

- Raw pixels (and simple functions of raw pixels)

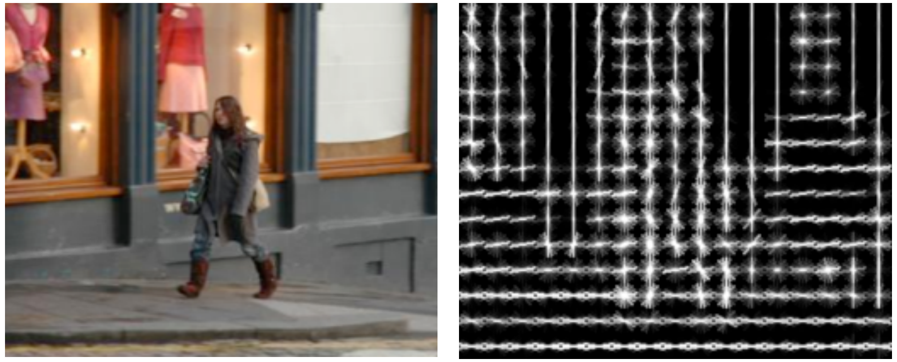
- Histograms, bags of features



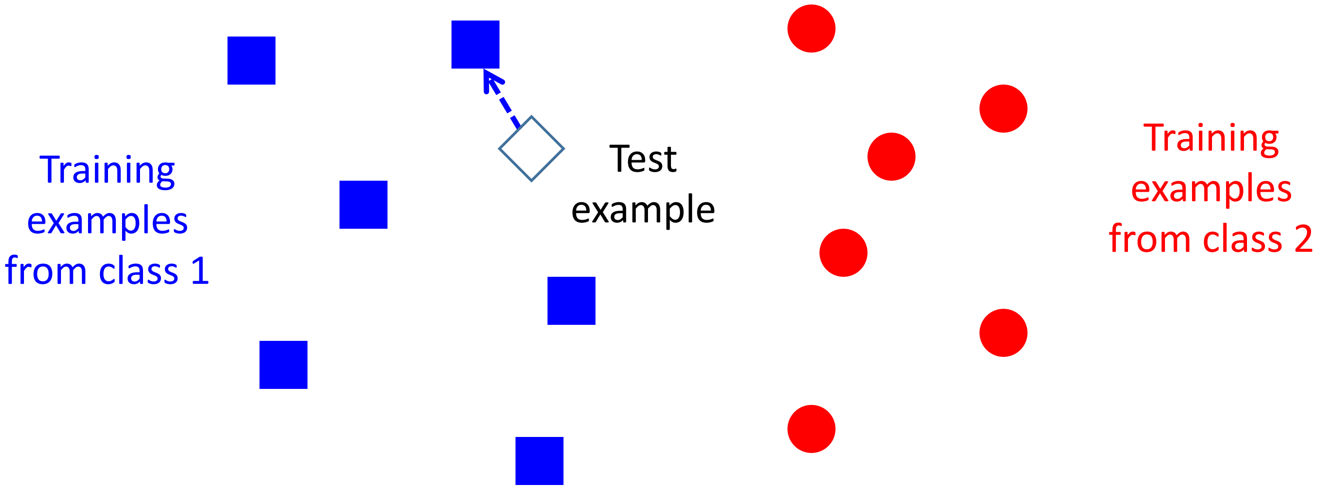
- GIST descriptors [Oliva and Torralba, 2001]



- Histograms of oriented gradients (HOG) [Dalal and Triggs, 2005]



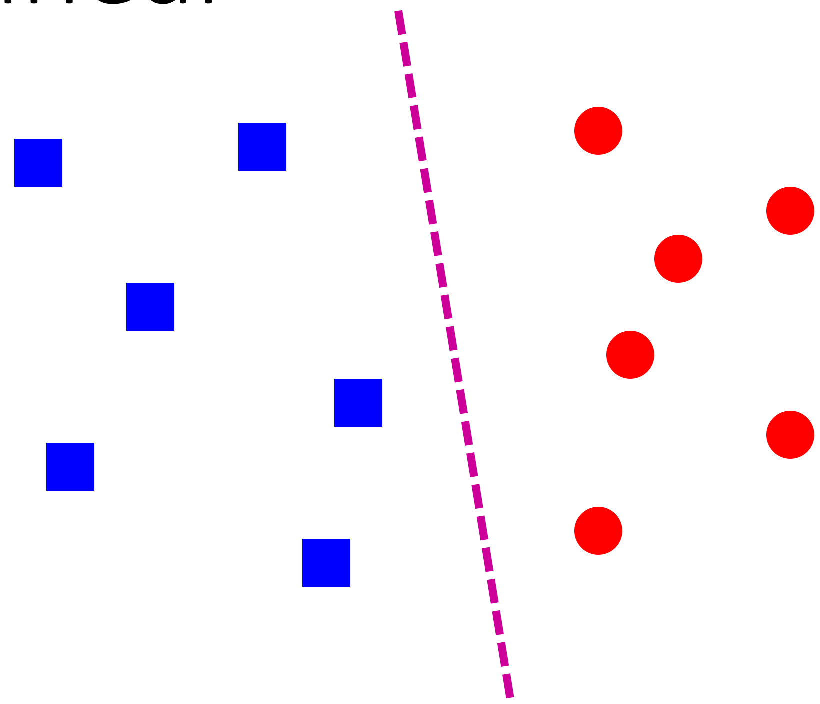
Classifiers: Nearest neighbor



f(**x**) = label of the training example nearest to **x**

- All we need is a distance function for our inputs
- No training required!

Classifiers: Linear



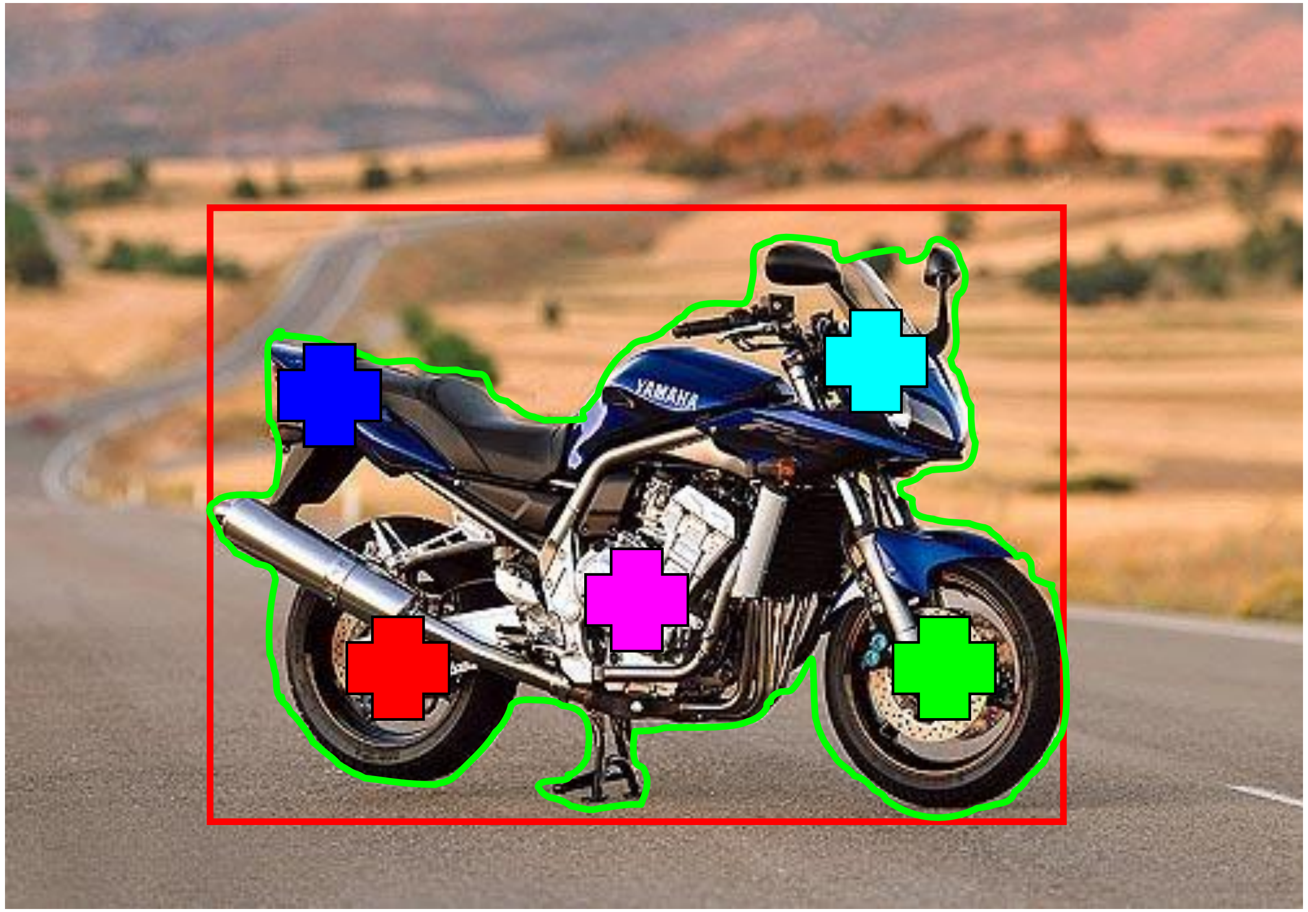
- Find a *linear function* to separate the classes:

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Recognition task and supervision

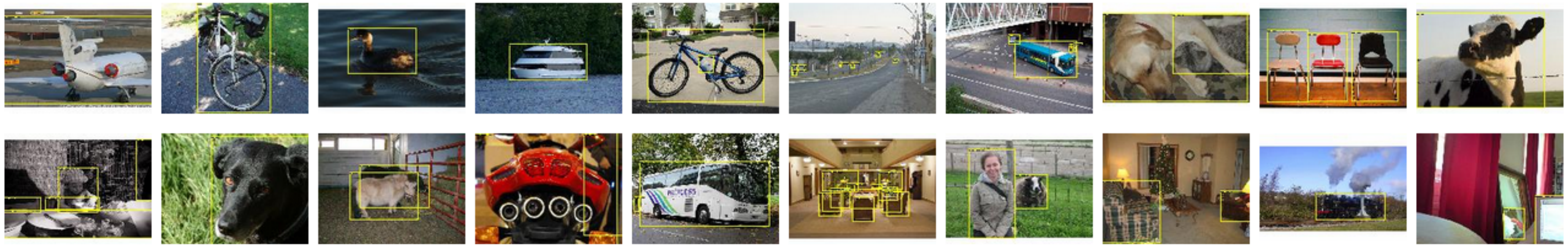
- Images in the training set must be annotated with the "correct answer" that the model is expected to produce

Contains a motorbike



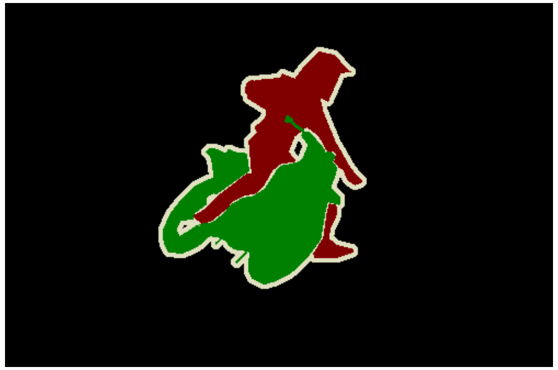
PASCAL competitions

- **Classification:** For each of the twenty classes, predicting presence/absence of an example of that class in the test image
- **Detection:** Predicting the bounding box and label of each object from the twenty target classes in the test image

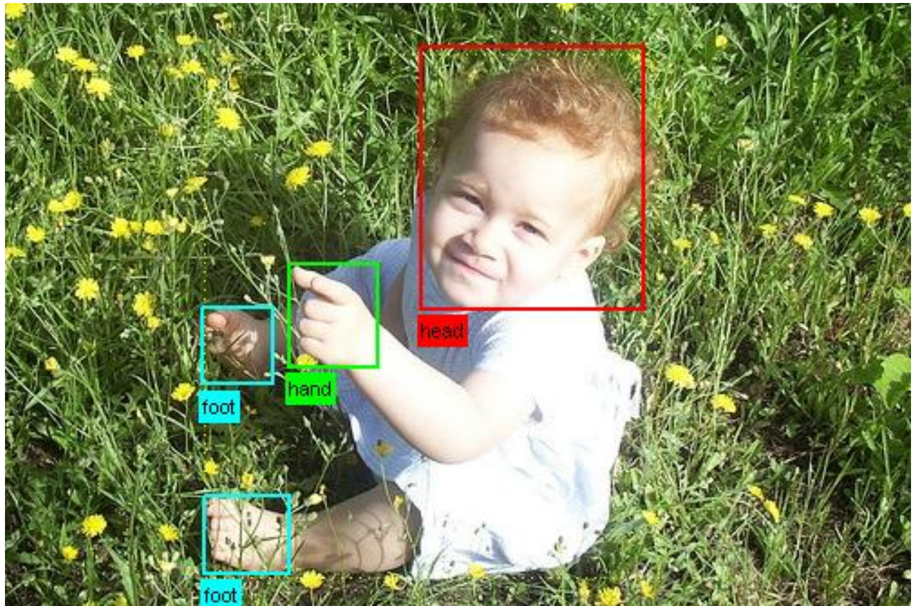


PASCAL competitions

- **Segmentation:** Generating pixel-wise segmentations giving the class of the object visible at each pixel, or "background" otherwise

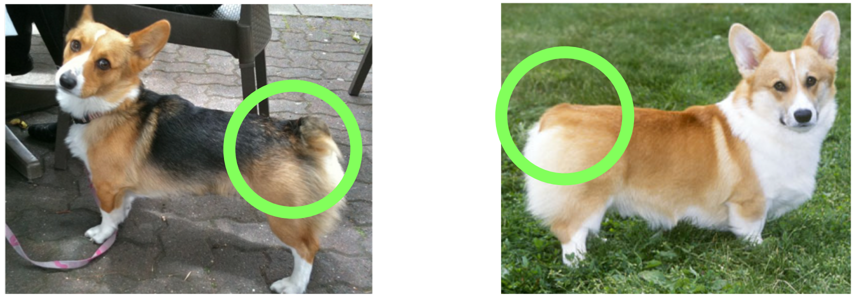


- **Person layout:** Predicting the bounding box and label of each part of a person (head, hands, feet)



Fine-grained recognition

Cardigan Welsh Corgi



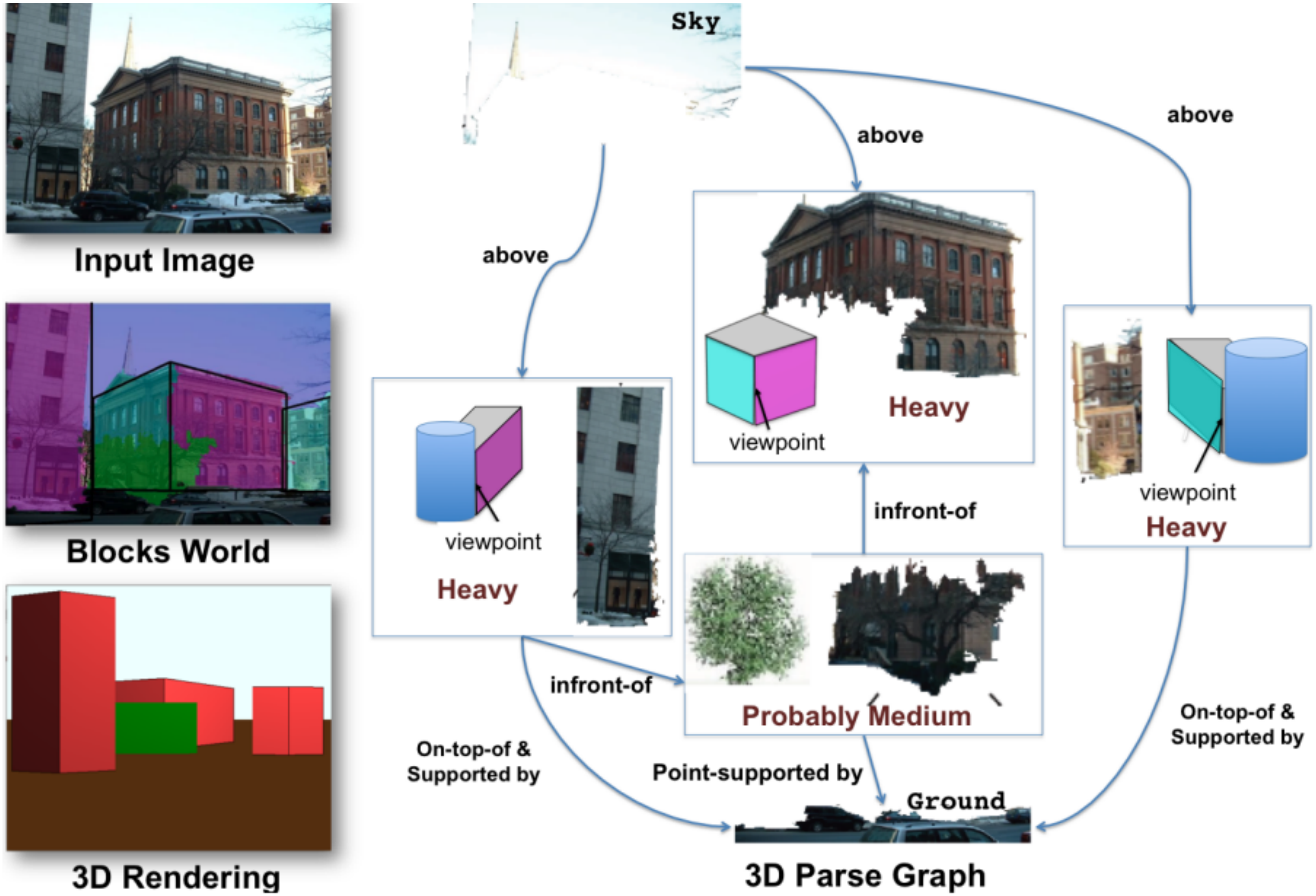
Pembroke Welsh Corgi

What breed is this dog?

Key: Find the right features.

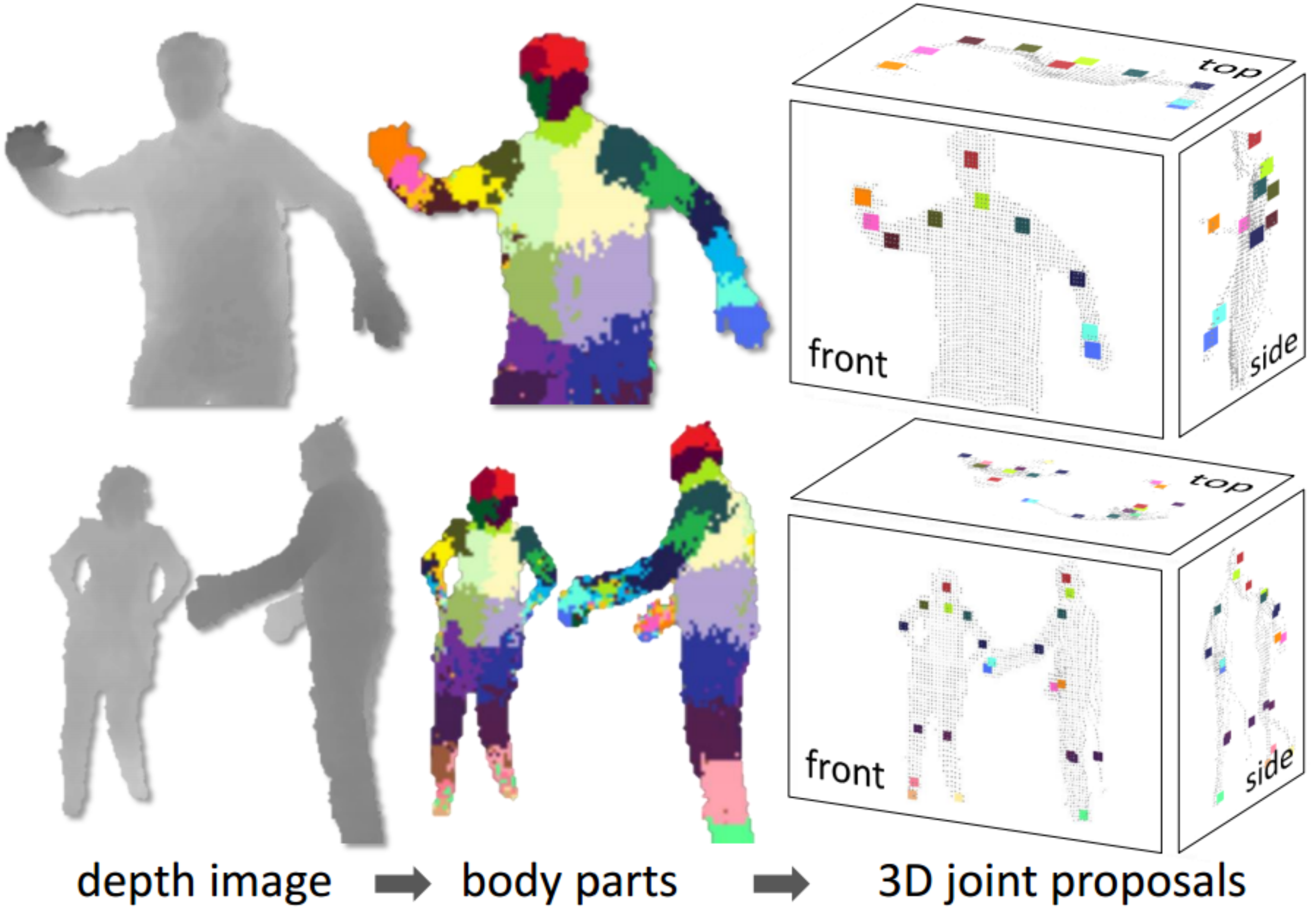
Source: J. Deng

Geometric image interpretation



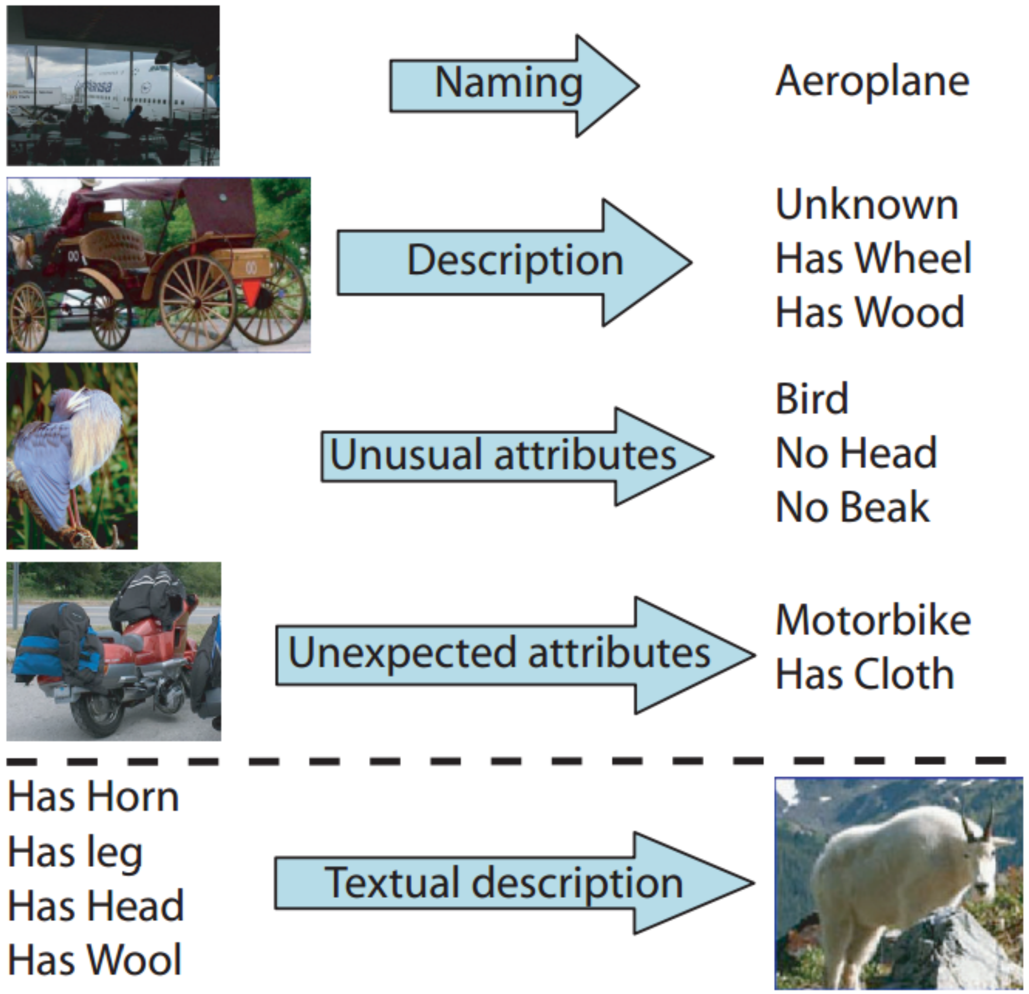
A. Gupta, A. Efros and M. Hebert, [Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics](), ECCV 2010

Recognition from RGBD Images



J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, [Real-Time Human Pose Recognition in Parts from a Single Depth Image](), CVPR 2011

Attribute-based recognition



A. Farhadi, I. Endres, D. Hoiem, and D Forsyth, [Describing Objects by their Attributes](), CVPR 2009