Summary of TEK5030

06.06.2019

Lectures 2019

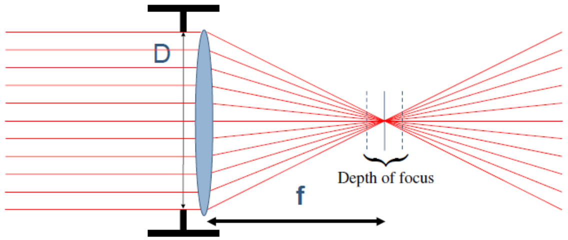
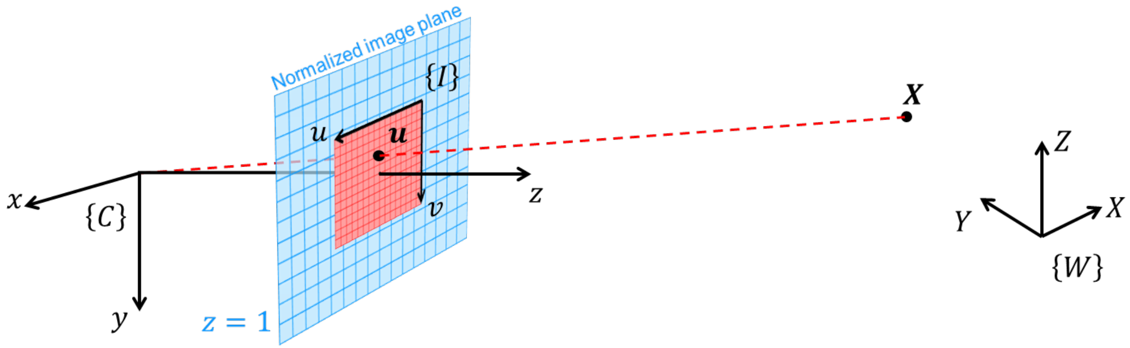
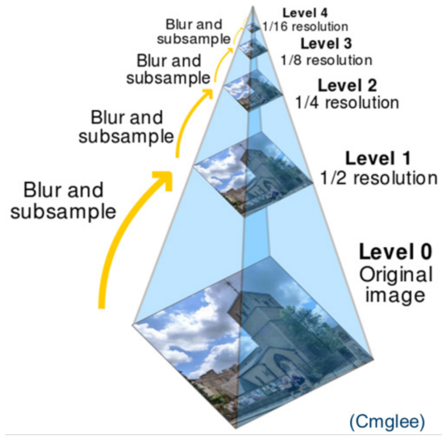
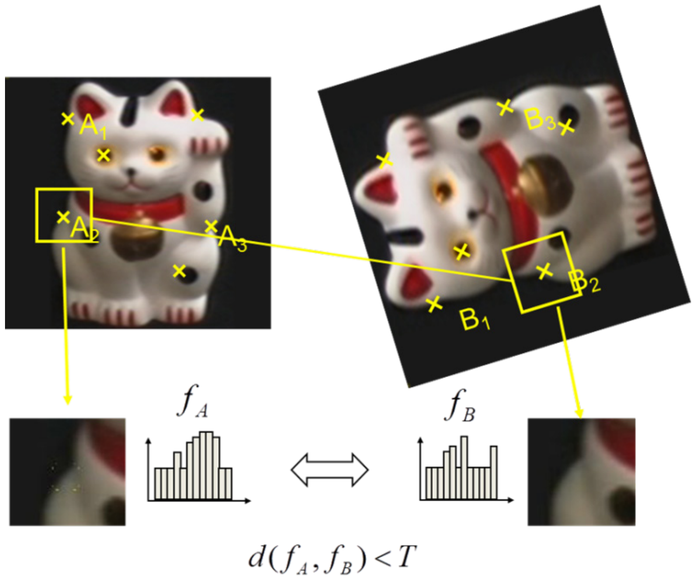


IMAGE FORMATION, PROCESSING AND FEATURES

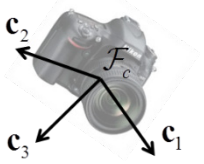
- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

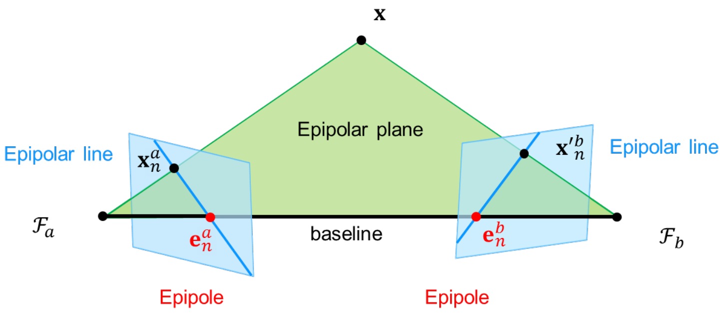


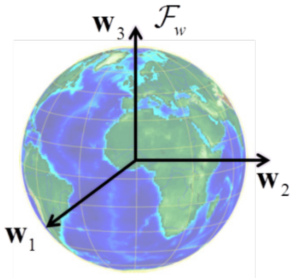




Lectures 2019

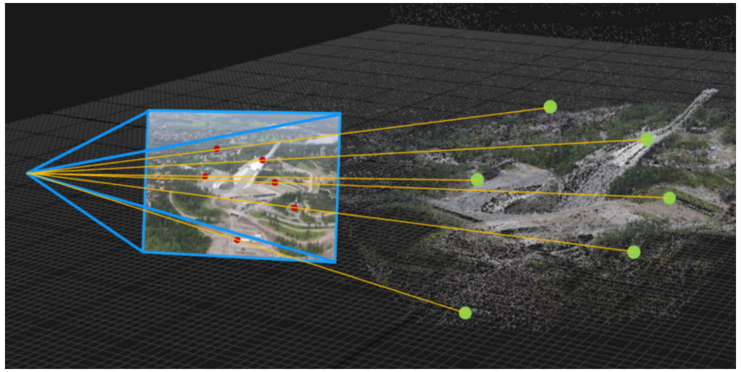


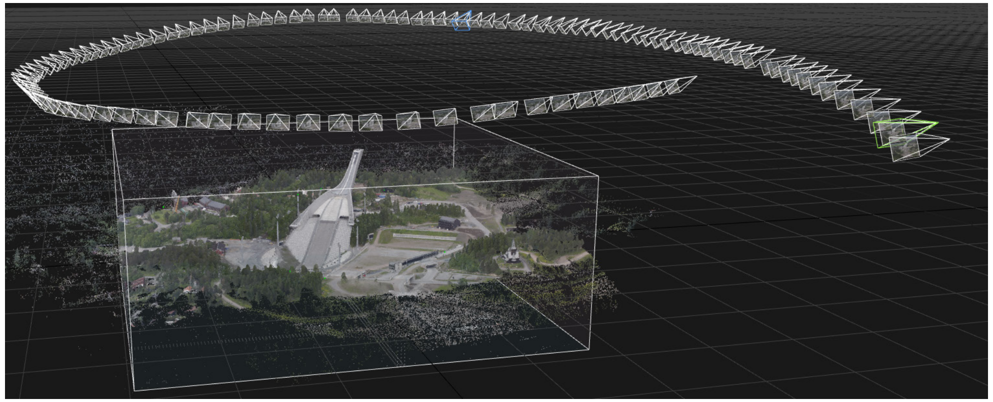




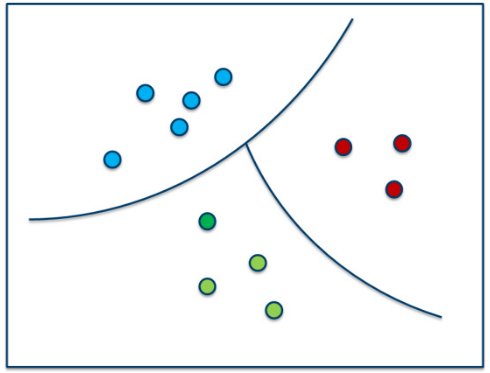
WORLD GEOMETRY AND 3D

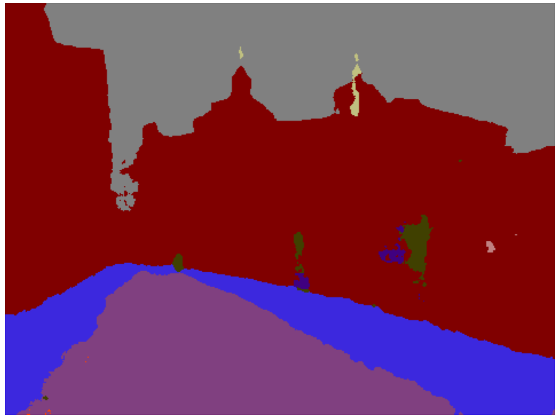
- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

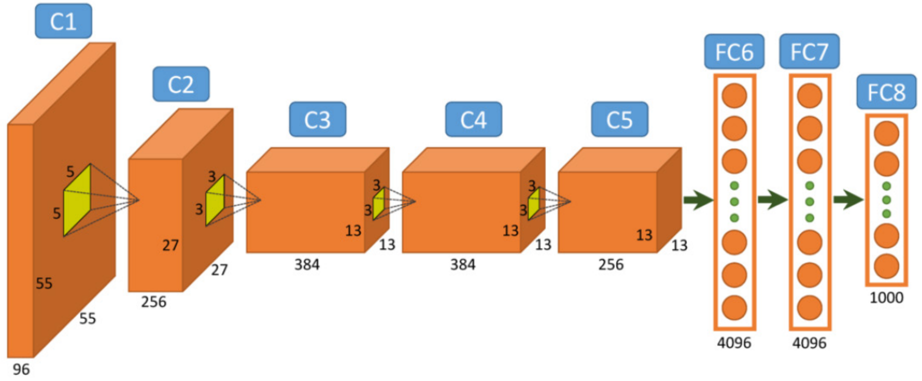


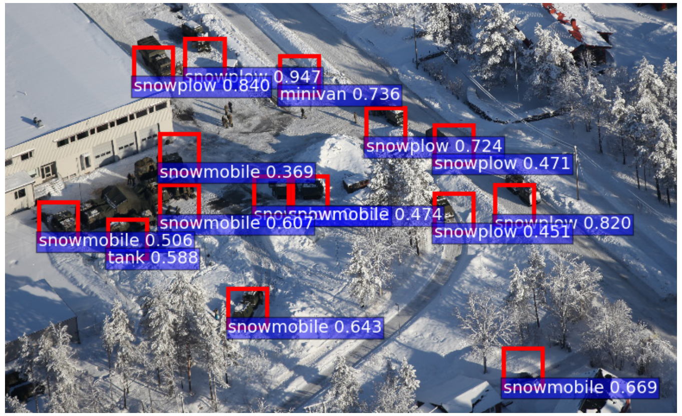


Lectures 2019









SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

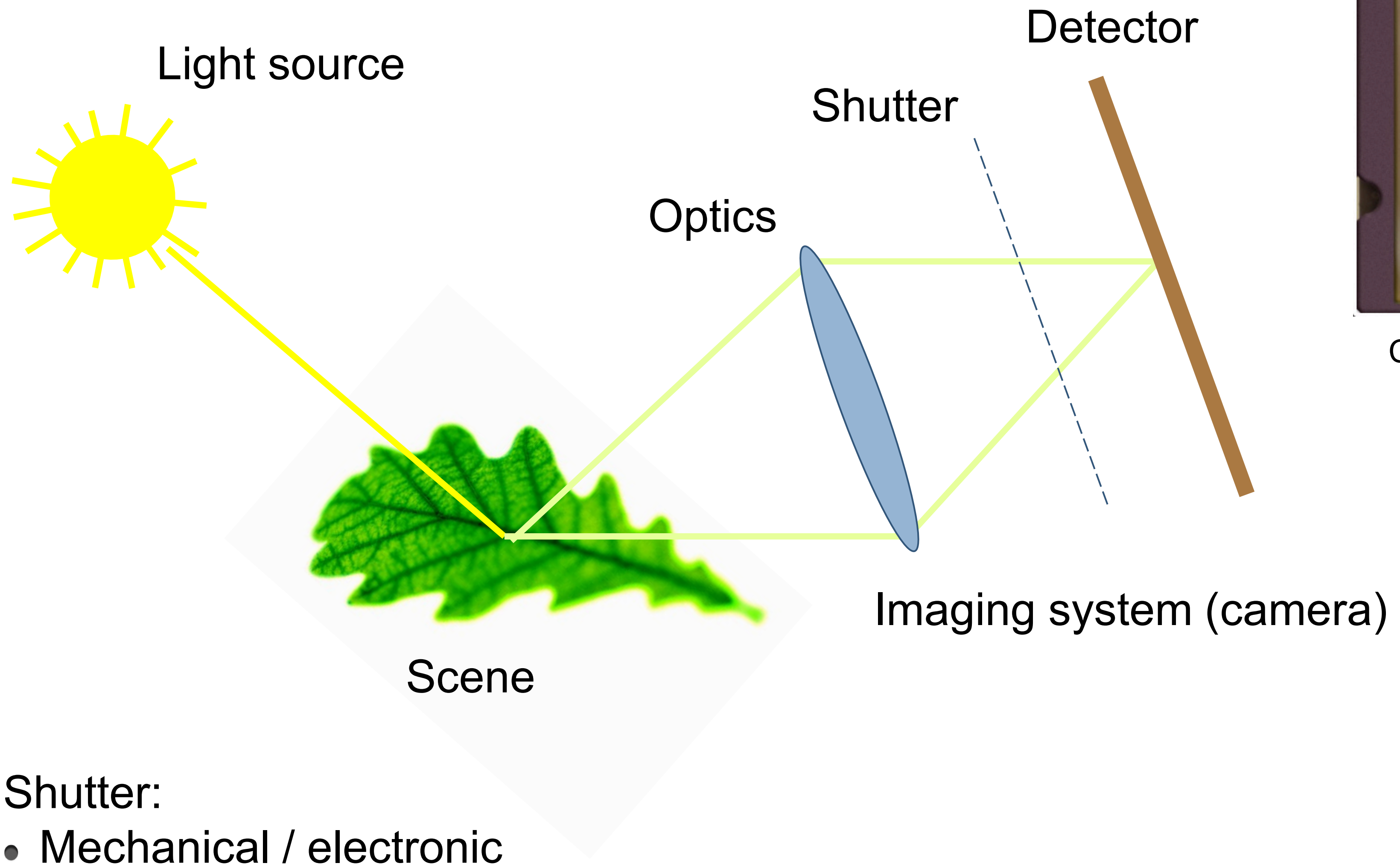
SCENE ANALYSIS

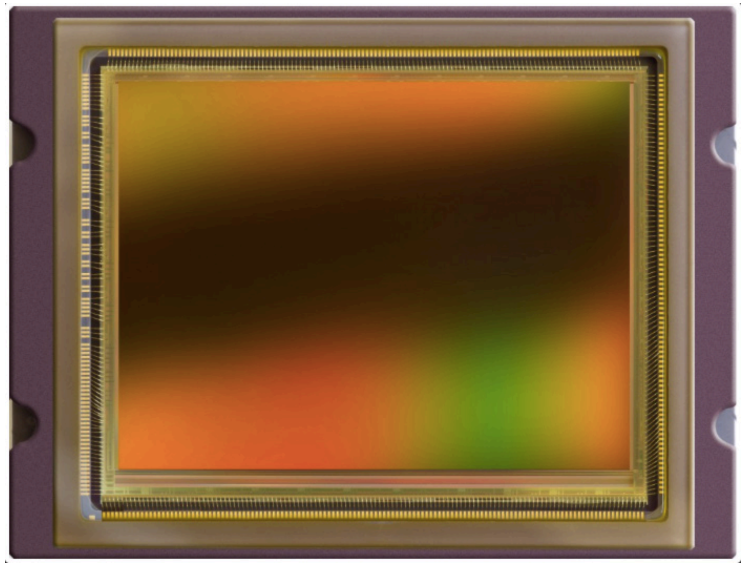
- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Image formation
Light, cameras, optics and colour

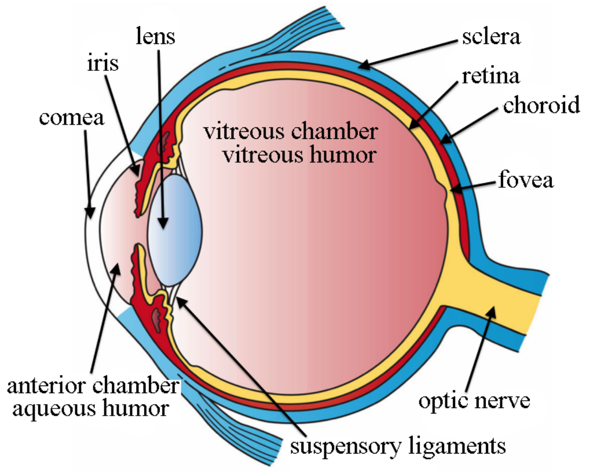
Image formation:
- Illumination
- Cameras
- Optics
- Colour Sensing.

Image capture





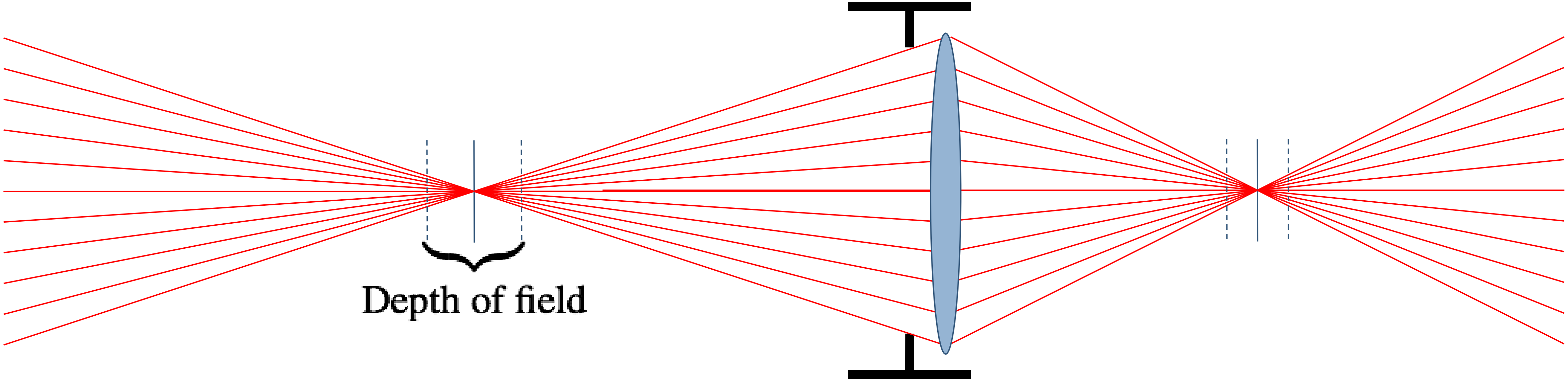
CMOS image sensor (CMOSIS 48Mp)



(Artwork by Holly Fischer)

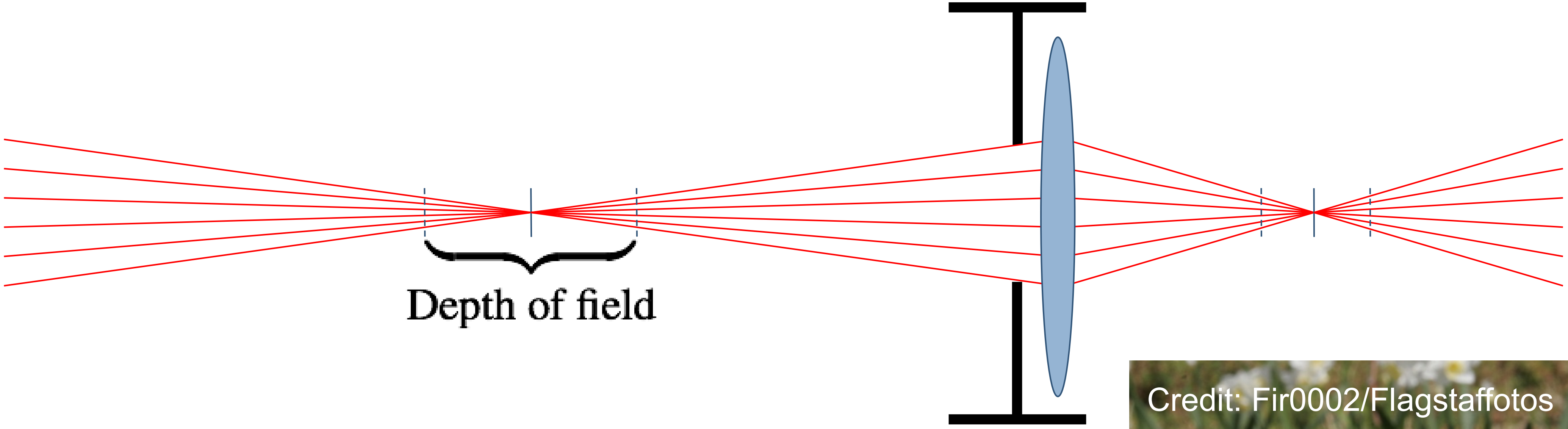
Shutter:
- Mechanical / electronic
- Global / rolling

Depth of field – large aperture



Large aperture → Narrow depth of field

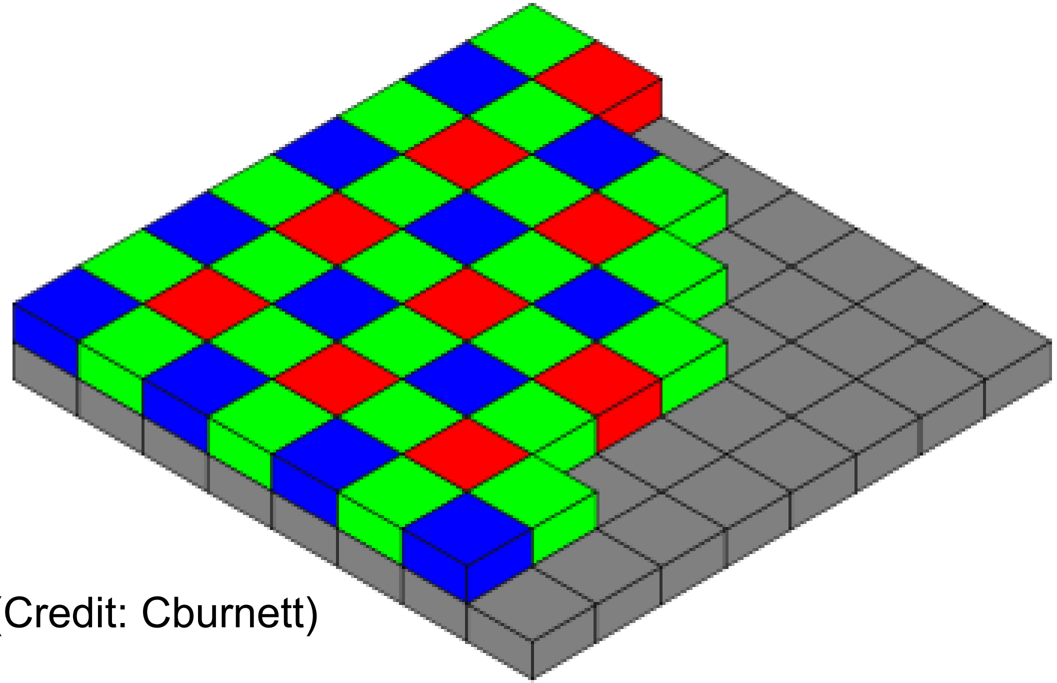
Depth of field – small aperture



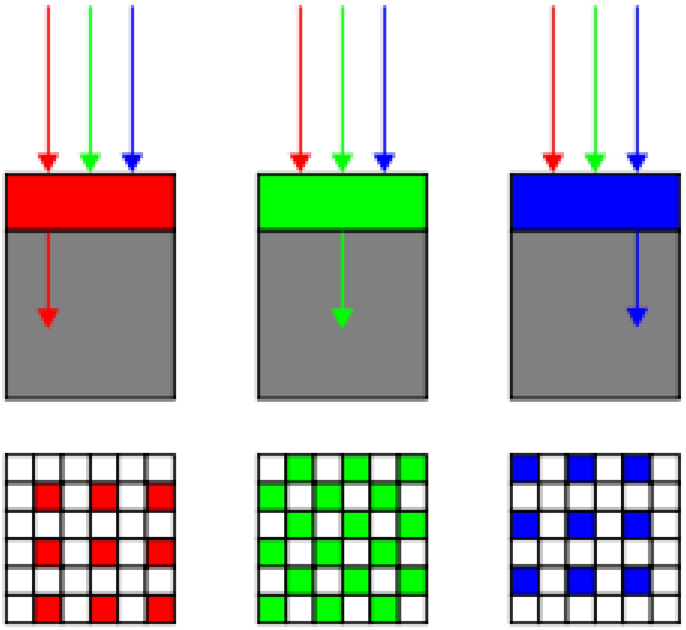
Small aperture → Large depth of field

Too small aperture will lead to *diffraction* and loss of sharpness

Colour Sensing in digital cameras - Bayer filter

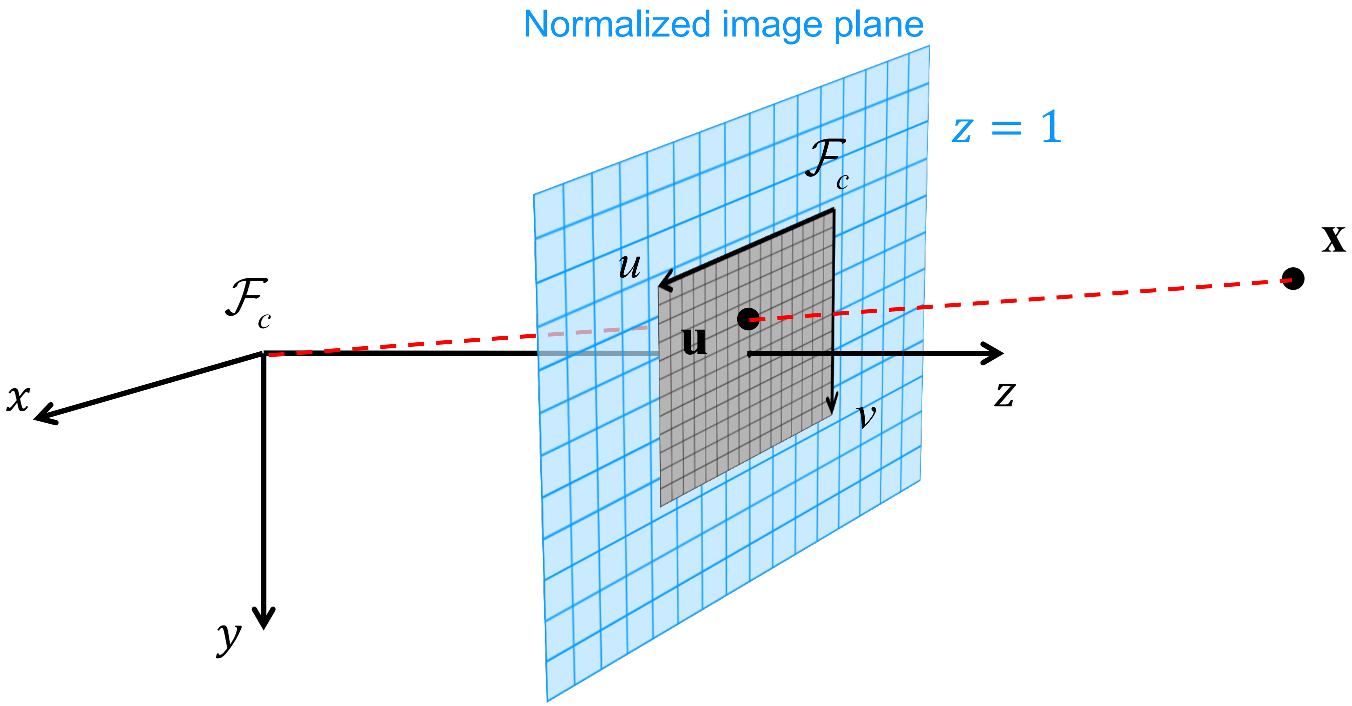


(Credit: Cburnett)



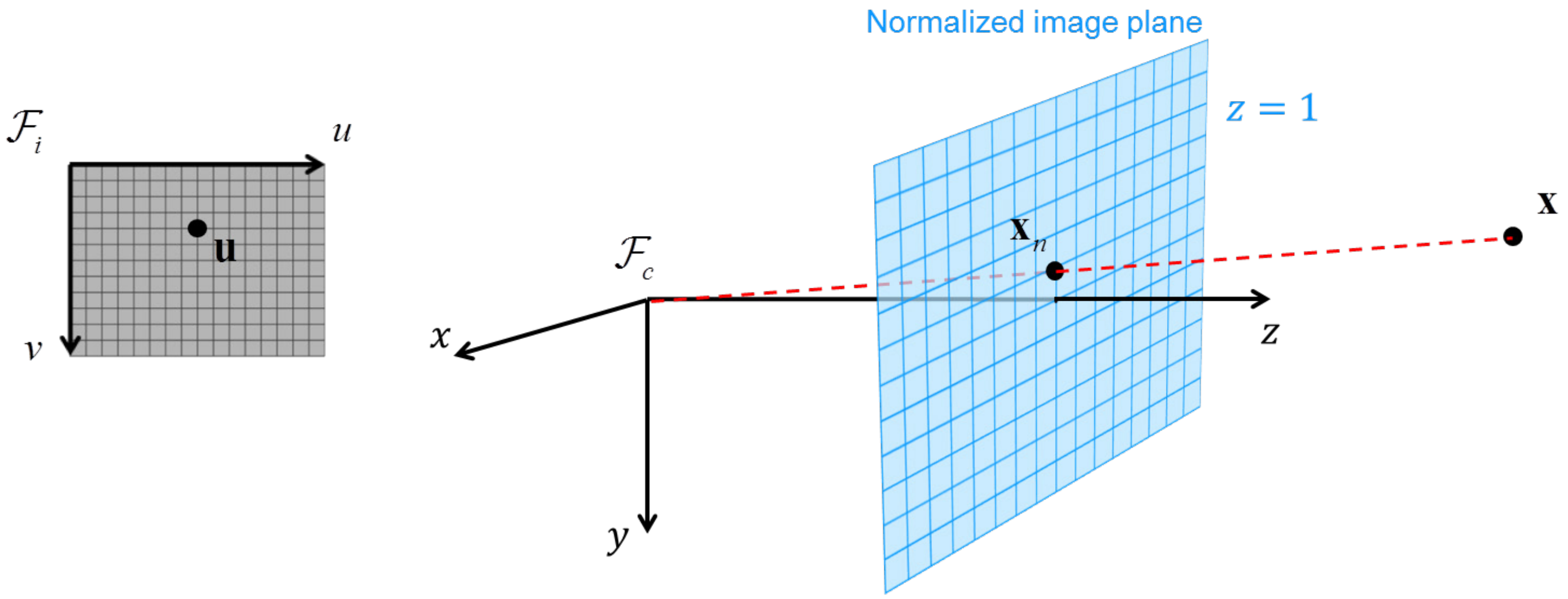
Undersampled (incomplete) colour information

The perspective camera model



The image is represented by a 2D frame $\mathcal{F}_i$ that spans the normalized image plane

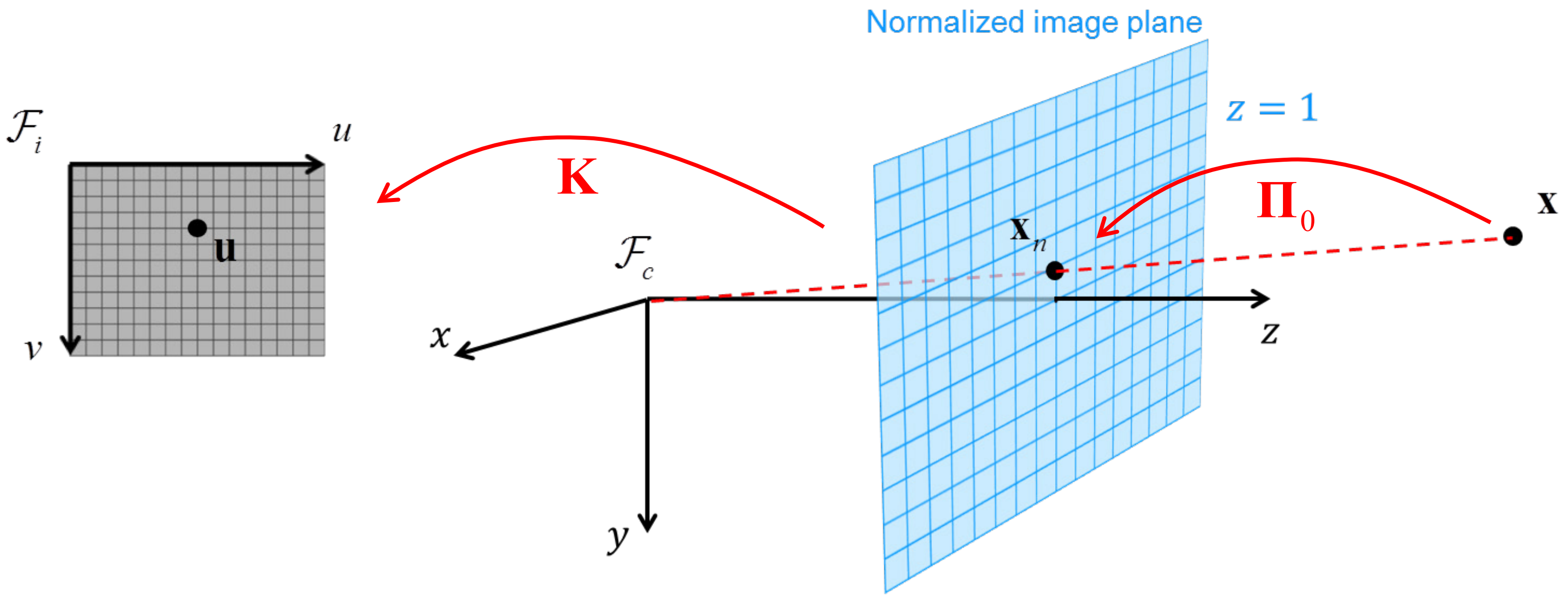
The perspective camera model



Points in the normalized image plane can be described both as 2D and 3D points

- 3D points $\mathbf{x}_n$ in $\mathcal{F}_c$
- 2D points $\mathbf{u}$ in $\mathcal{F}_i$

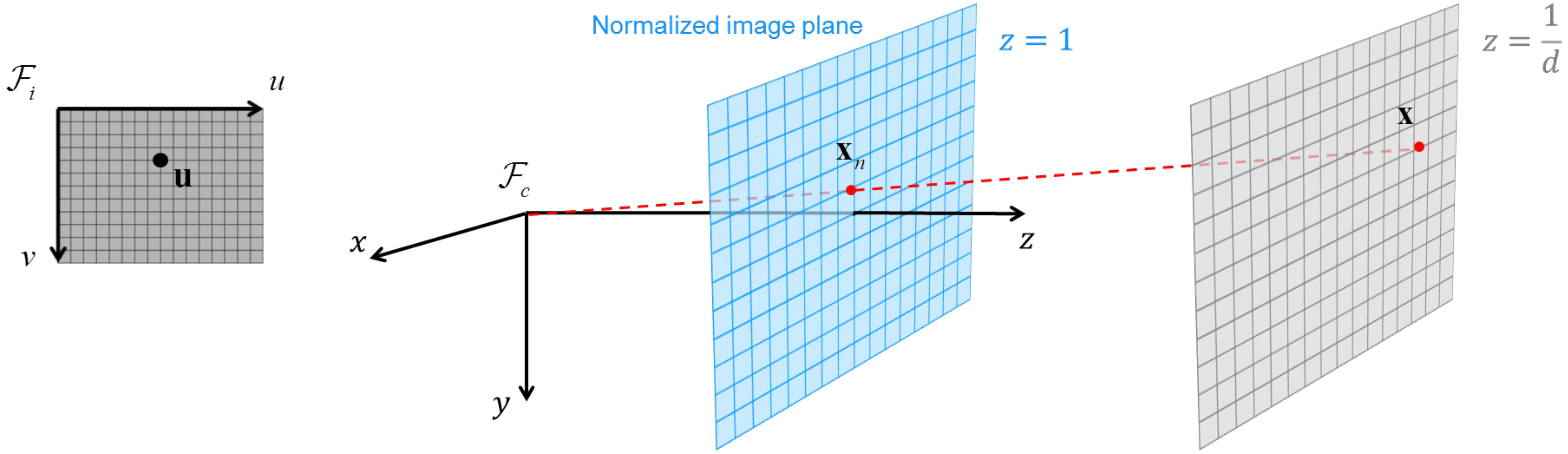
The perspective camera model



The perspective camera model is composed by two transformations:

$$\tilde{\mathbf{u}} = \underbrace{\begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{\Pi}_0} \tilde{\mathbf{x}}$$

Inverting the perspective camera model



$$\mathbf{x} = \frac{1}{d}\mathbf{K}^{-1}\begin{bmatrix}\mathbf{u} \\ 1\end{bmatrix}$$

For a given inverse depth $d = \frac{1}{z}$

Remark on computations

Computing the image point $\mathbf{u} = [u, v]^T$ for a world point $\mathbf{x} = [x, y, z]^T$ can be split into three steps

$$\mathbf{x} \mapsto \tilde{\mathbf{x}} \mapsto \tilde{\mathbf{u}} = \mathbf{K}\mathbf{\Pi}_0\tilde{\mathbf{x}} \mapsto \mathbf{u}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \qquad \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\tilde{u}}{\tilde{w}} \\ \frac{\tilde{v}}{\tilde{w}} \end{bmatrix}$$

Homogeneous coordinates!

The camera calibration matrix

$$\mathbf{K} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}$$

- This is an affine transformation from the normalized image plane to the image

$$\tilde{\mathbf{u}} = \mathbf{K}\tilde{\mathbf{x}}_n$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$u = f_u x + sy + c_u$$

$$v = f_v y + c_v$$

- The **principal point**, (c_u, c_v) is where the optical axis intersects the image plane
  - Often approximated by the center of the image
- The **focal lengths** f_{u} and f_v are scale factors between the normalized image plane and the image
  - They are scaled versions of the physical focal length
- The **skew parameter** s can typically be ignored, so we usually set $s = 0$
  - It is required for cases when the detector array is not orthogonal to the optical axis

Non-ideal cameras

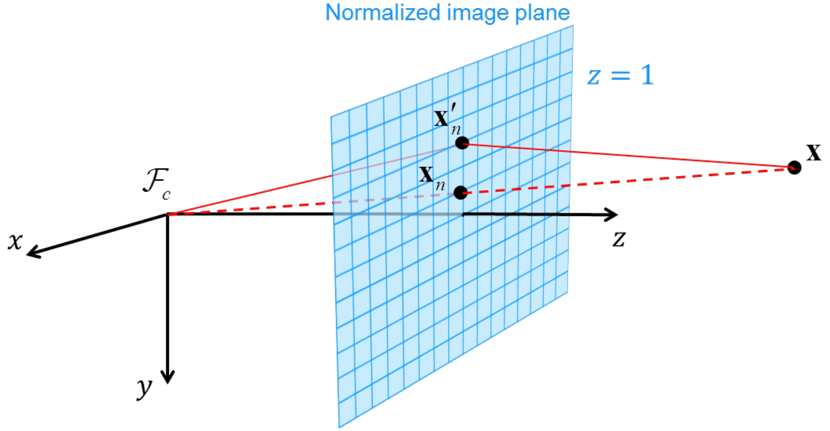
- No cameras fit the perspective camera model perfectly
  - All cameras suffer from some kind of distortion

- If we want to use images for geometrical computations we need to take this distortion into account

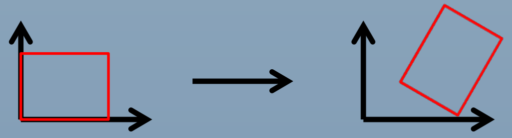
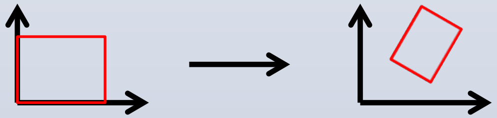
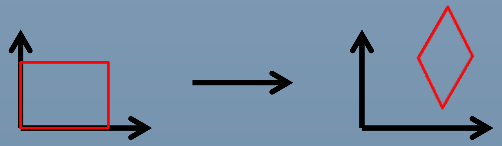
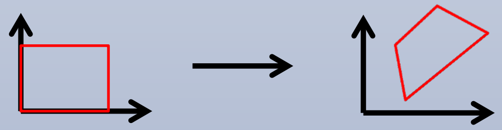
- A **distortion model** allows us to undistort images (or individual points)
  - Example model for radial distortion only

$$x_n = x'_n \left(1 + k_1 r'^2 + k_2 r'^4\right)$$
$$y_n = y'_n \left(1 + k_1 r'^2 + k_2 r'^4\right)$$

where $r'^2 = x'^2_n + y'^2_n$



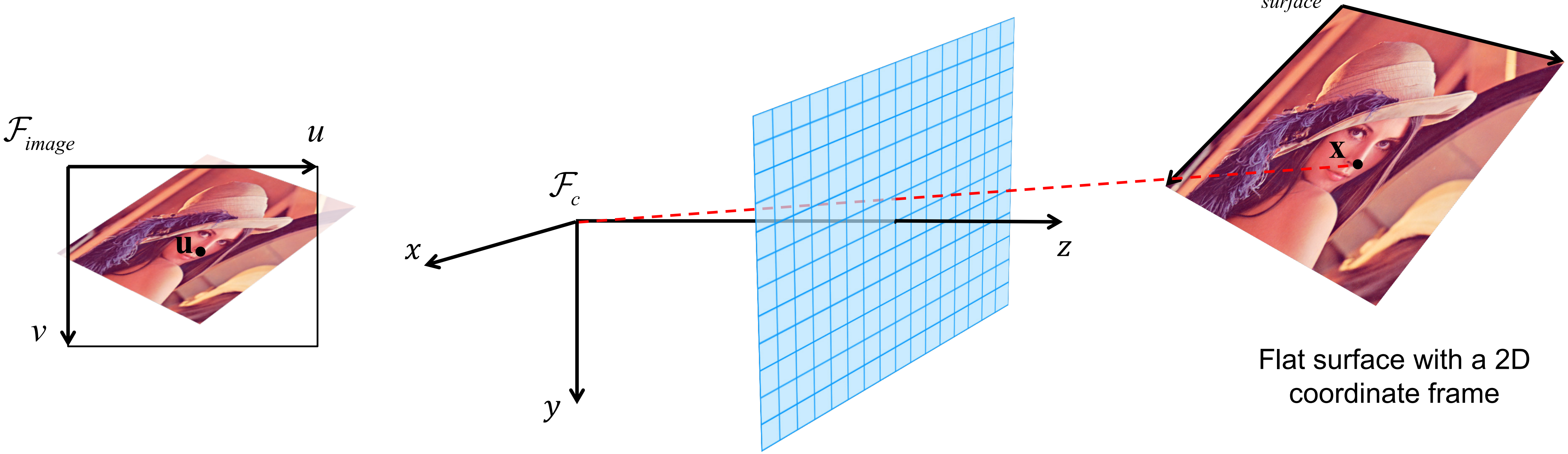
Linear transformations of the projective plane $\mathbb{P}^2$

Transformation	Matrix	#DoF	Preserves	Visualization
Euclidean	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	3	Lengths **+ all below**	
Similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad s \in \mathbb{R}$	4	Angles **+ all below**	
Affine	$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$	6	Parallelism, line at infinity **+ all below**	
Homography	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$	8	Straight lines	

Linear transformations of the projective plane $\mathbb{P}^2$

- Perspective imaging of a flat surface can be described by a homography

$$\mathbf{H}\tilde{\mathbf{x}}^{surface} = \tilde{\mathbf{u}}^{image}$$



Flat surface with a 2D coordinate frame

Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Image processing

- Point operators (pixel-to-pixel)
  - Adjustment of brightness, contrast and colour
  - Histogram equalization

- Image filtering in spatial domain
  - Mathematical operations on a local neighborhood
  - Linear filters (convolution, cross-correlation)
  - Non-linear filters
  - Image enhancement (smoothing, sharpening)
  - Feature extraction (edges, texture etc.)

- Image filtering in frequency domain
  - Modification of spatial image frequencies
  - Noise removal, (re)sampling, image compression
  - 2D Fourier transform

$$f[i,j] \rightarrow g[i,j]$$

Linear filtering (cross-correlation or convolution)

f

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	0	0	0	0	0
0	0	0	90	90	90	0	0	0	0
0	0	90	90	90	90	0	0	0	0
0	0	90	90	90	90	90	0	0	0
0	90	90	90	90	90	90	90	0	0
0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\otimes$ $\frac{1}{9}$ h

1	1	1
1	1	1
1	1	1

$=$ g

0	0	0	0	0	0	0	0	0	0
0	0	10	20	20	10	0	0	0	0
0	0	20							

$$g[i,j] = \sum_{u,v} h[u,v] f[i+u, j+v]$$

$$g = h \otimes f$$

Filtering in frequency domain

Fourier (1807):

> ***Any*** *univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies (true with some subtle restrictions).*

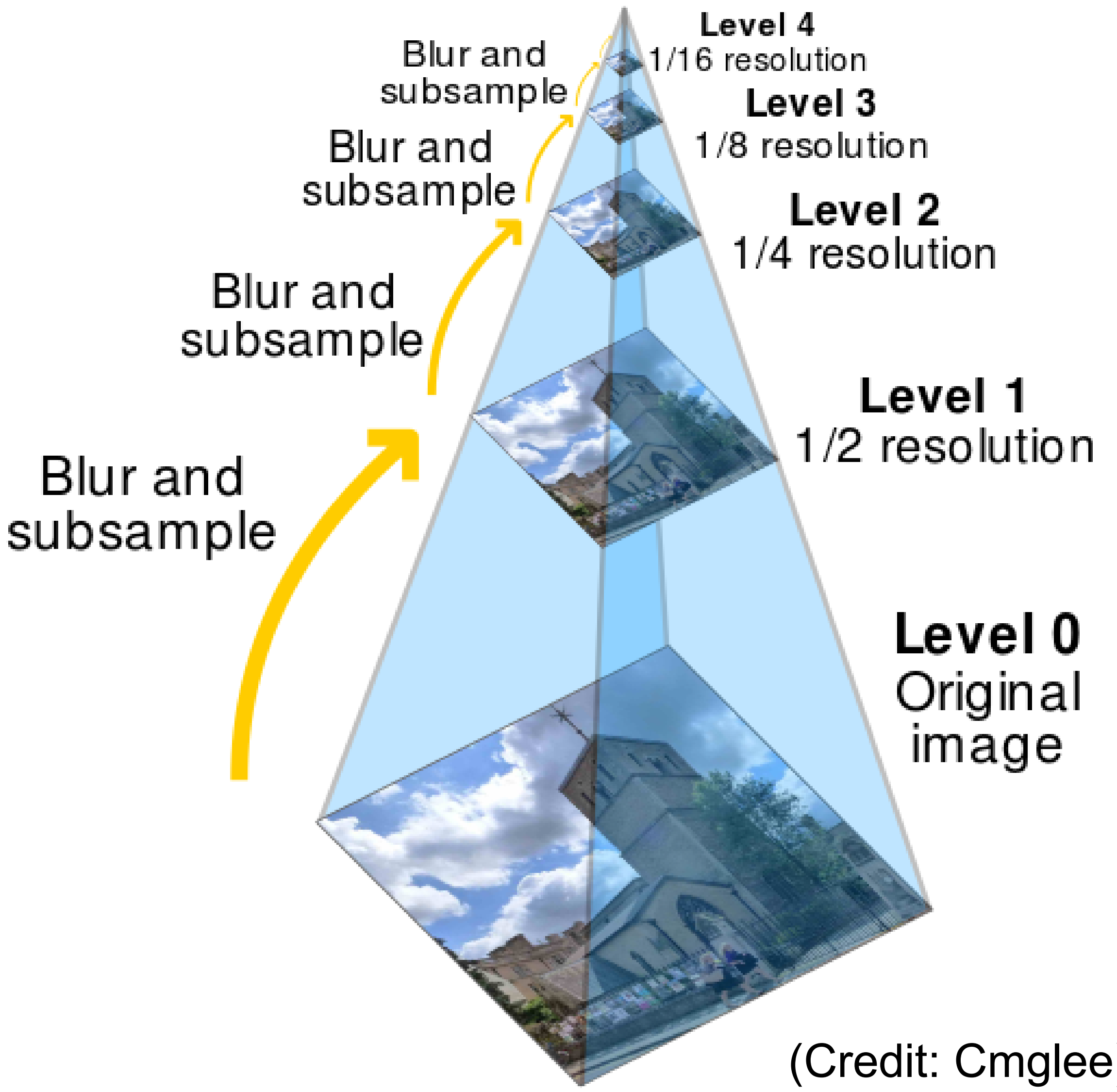
This leads to:

- Fourier Series
- Fourier Transform (continuous and discrete)
- Fast Fourier Transform (FFT)

Jean Baptiste Joseph Fourier (1768-1830)

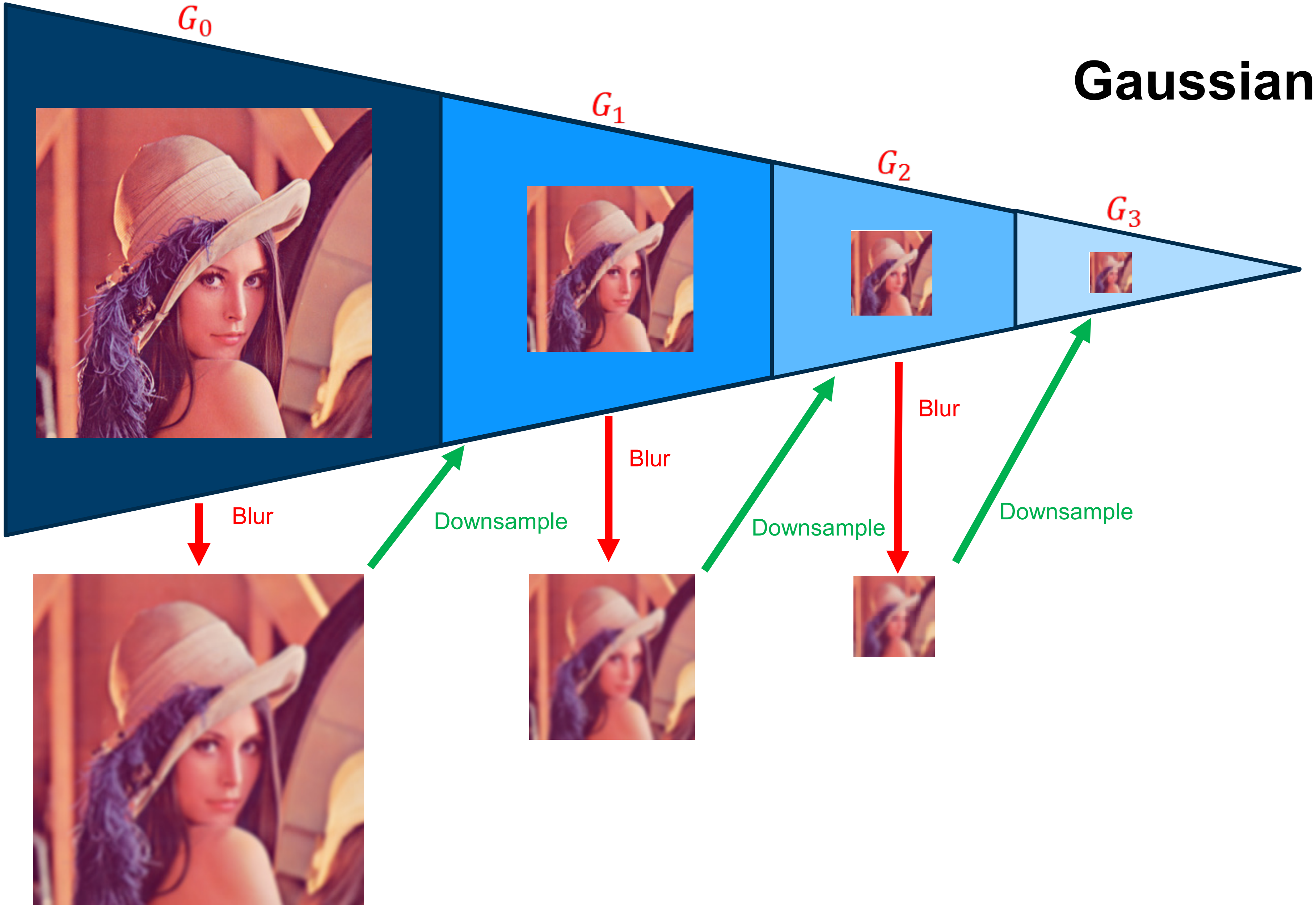
Image Pyramids

- Downsampling (decimation)
- Upsampling (interpolation)
- Pyramids
  - Gaussan Pyramids
  - Laplacian Pyramids
- Applications
  - Template matching (object detection)
  - Detecting stable points of interest
  - Image Registration
  - Compression
  - **Image Blending**
  - …

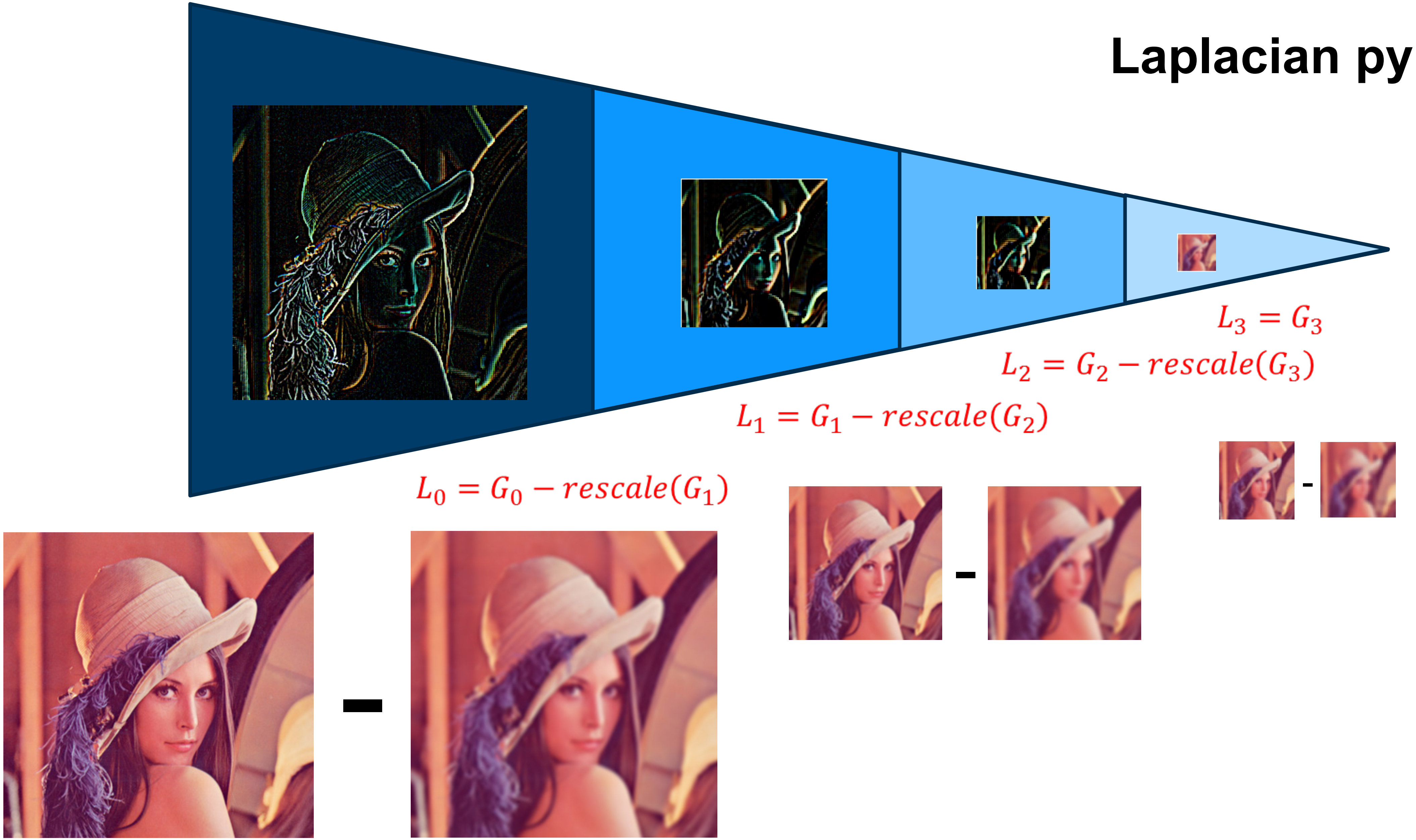


(Credit: Cmglee)

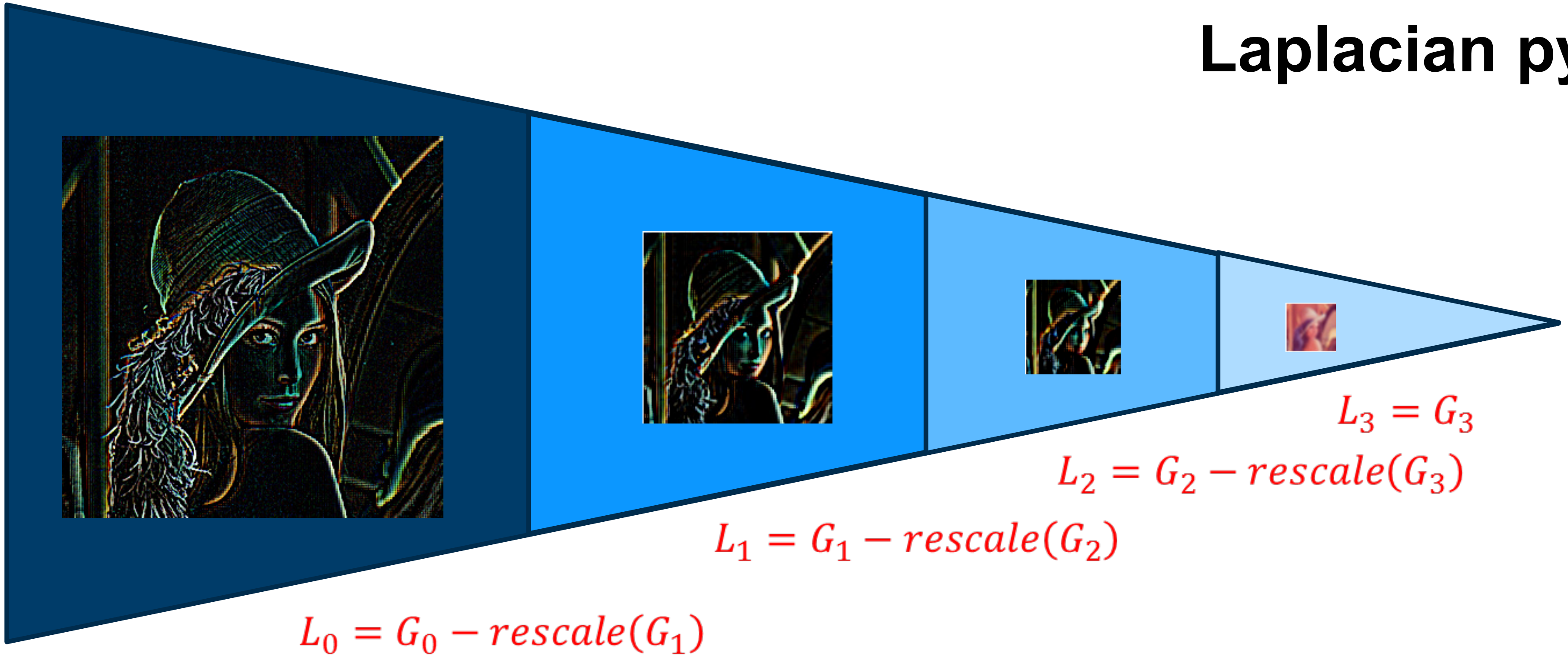
Gaussian Pyramid



Laplacian pyramid



Laplacian pyramid



$L_3 = G_3$

$L_2 = G_2 - rescale(G_3)$

$L_1 = G_1 - rescale(G_2)$

$L_0 = G_0 - rescale(G_1)$

Collapsing the Laplacian pyramid:

$rescale(rescale(rescale(L_3) + L_2) + L_1) + L_0 =$

Image blending

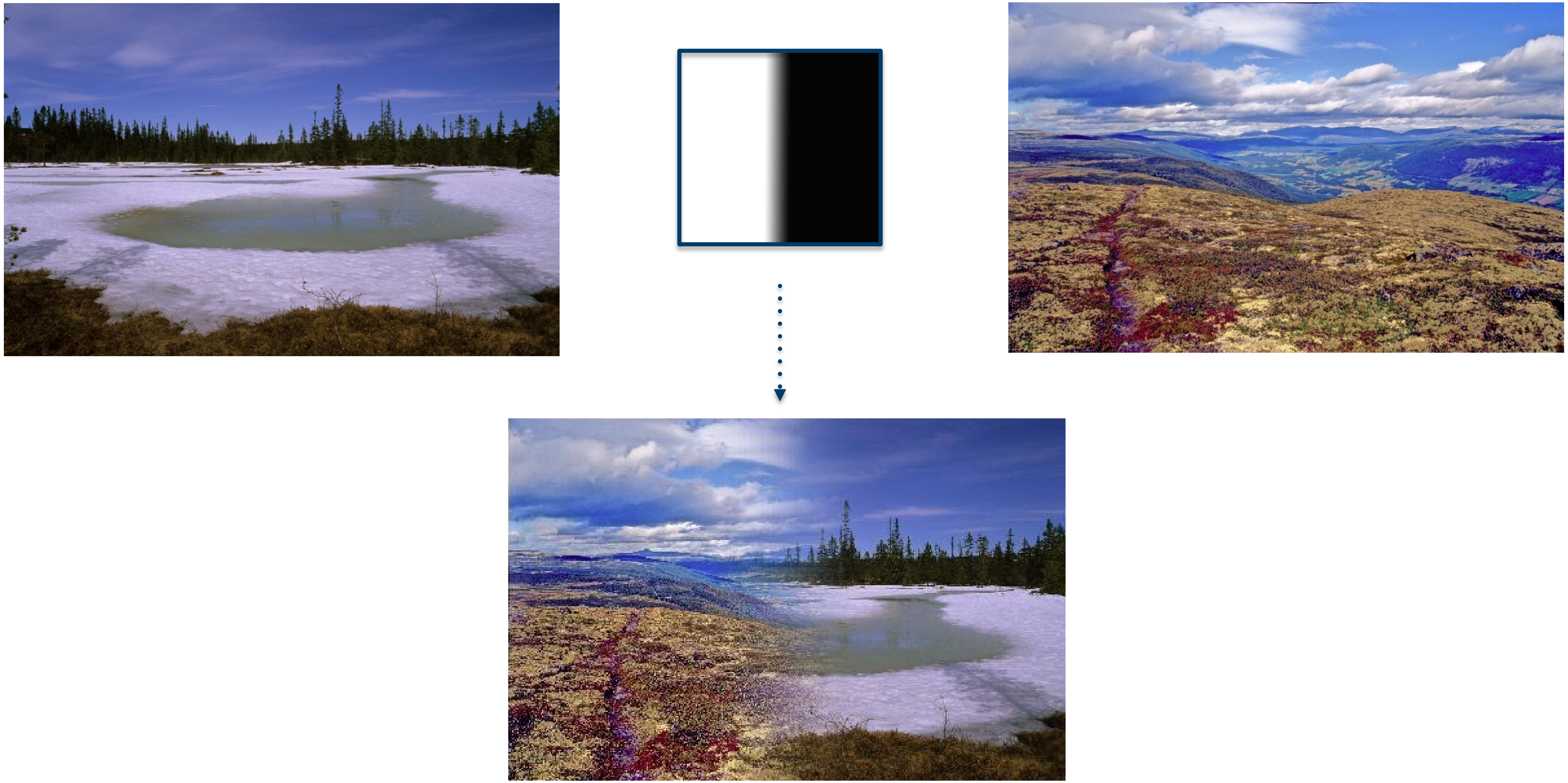
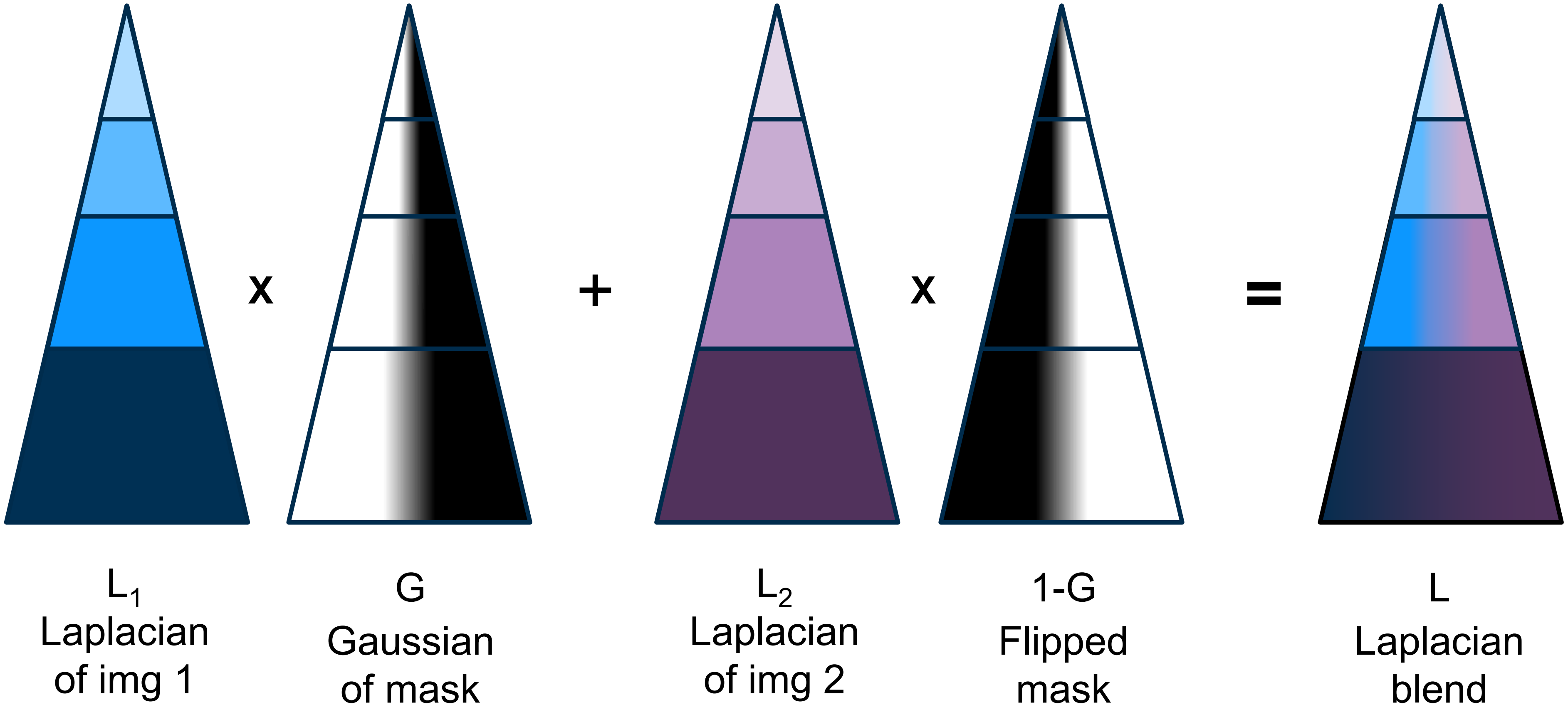


Image blending with Laplacian pyramids

Weighted sum for each level of the pyramid



Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

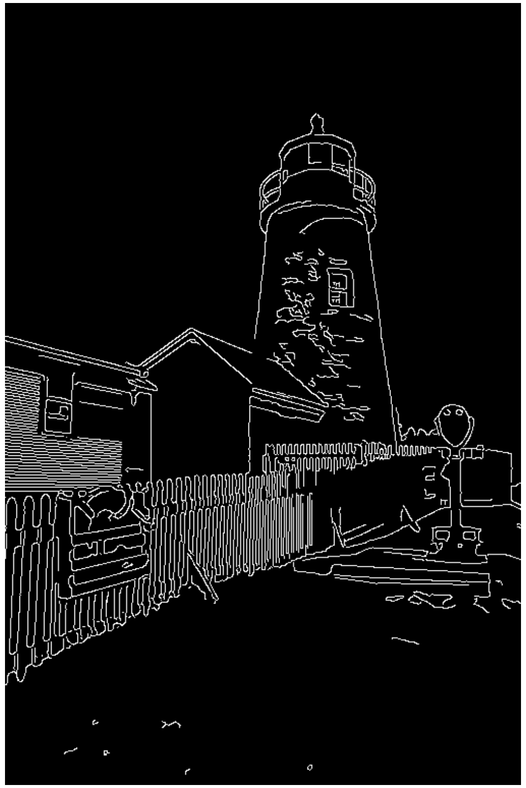
SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Feature detection

Line features:

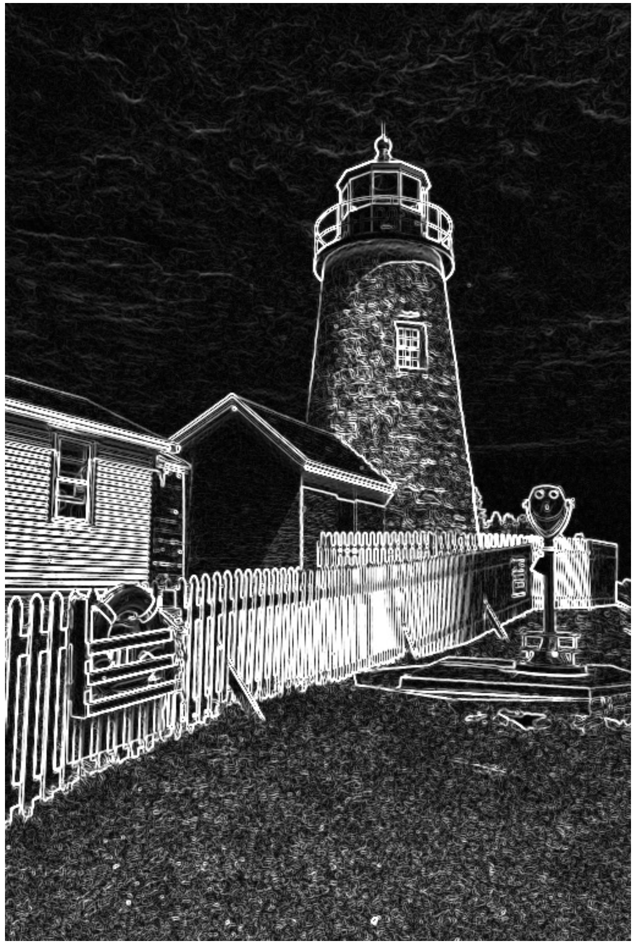
- Edge detectors
  - Image derivatives
  - Thinning and thresholding
- Line detection with the Hough transform



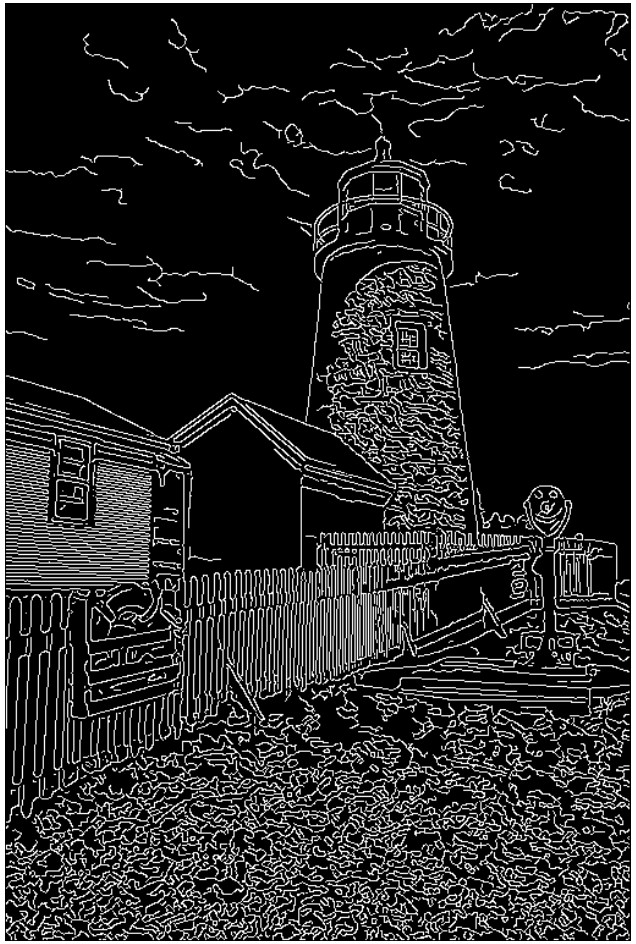
Thinning and thresholding

- Detection of local maxima (i.e. suppression of non-maxima)
- Thresholding

Binary image with isolated edges (single pixels at discrete locations along edge contours)



Edge enhanced image (Sobel)

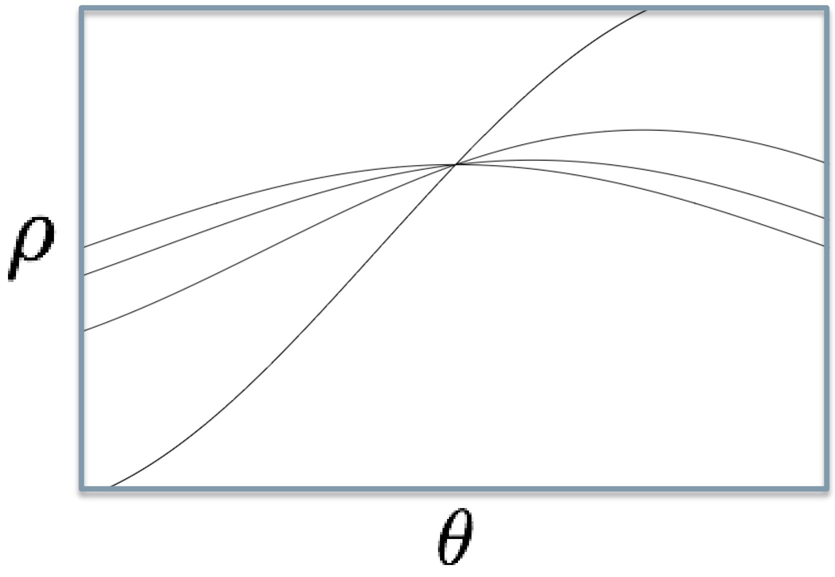


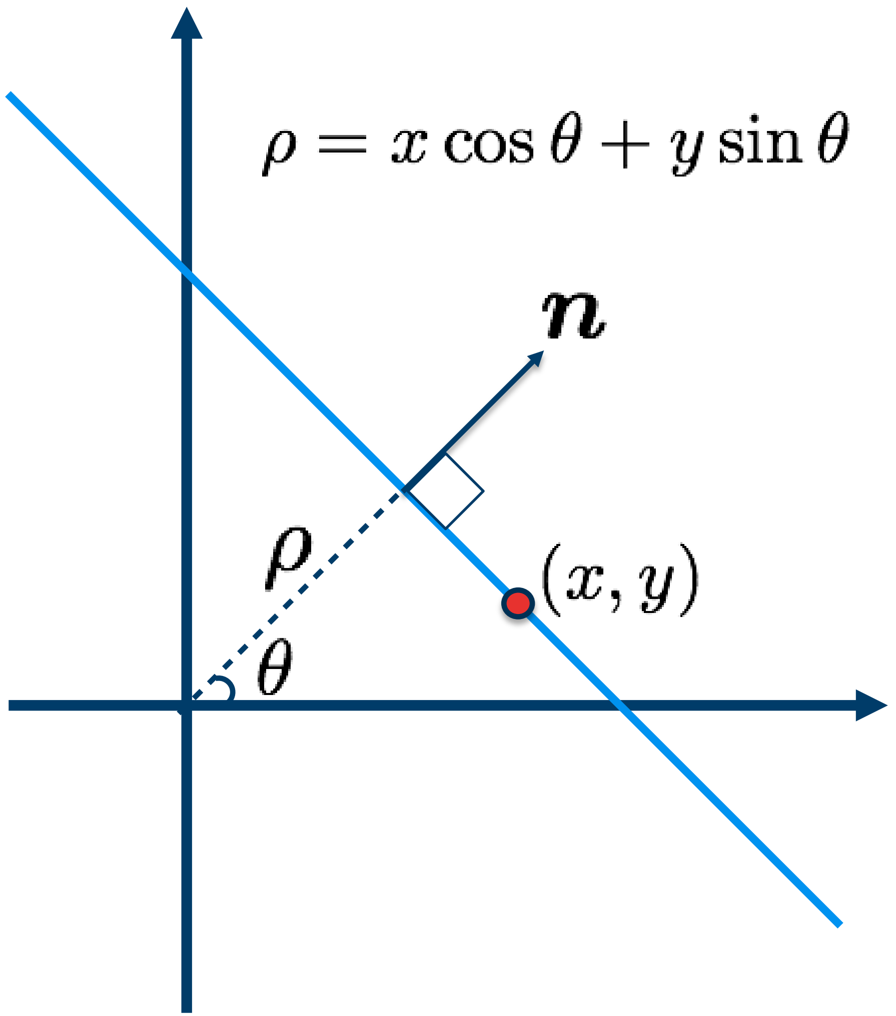
Edge image (Canny)

Line detection - Hough transform

The set of all lines going through a given point corresponds to a sinusoidal curve in the (ρ, θ) plane.

Two or more points on a straight line will give rise to sinusoids intersecting at the point (ρ, θ) for that line.

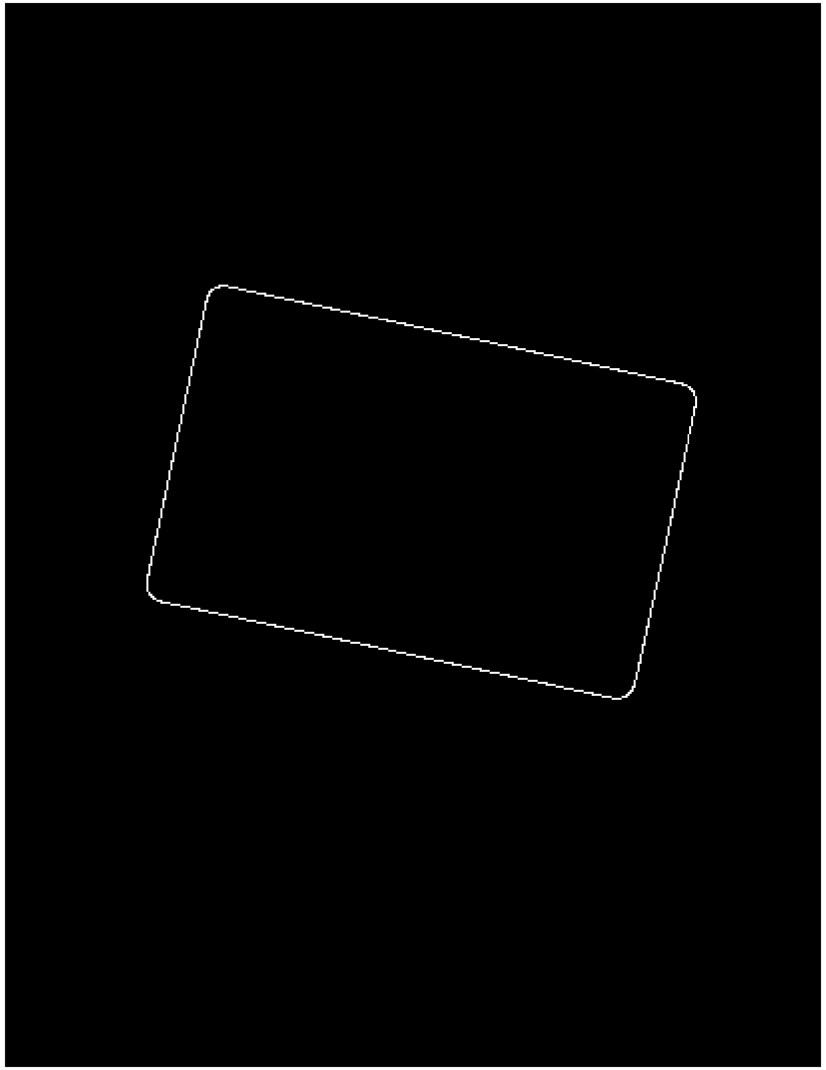




The Hough transform can be generalized to other shapes.

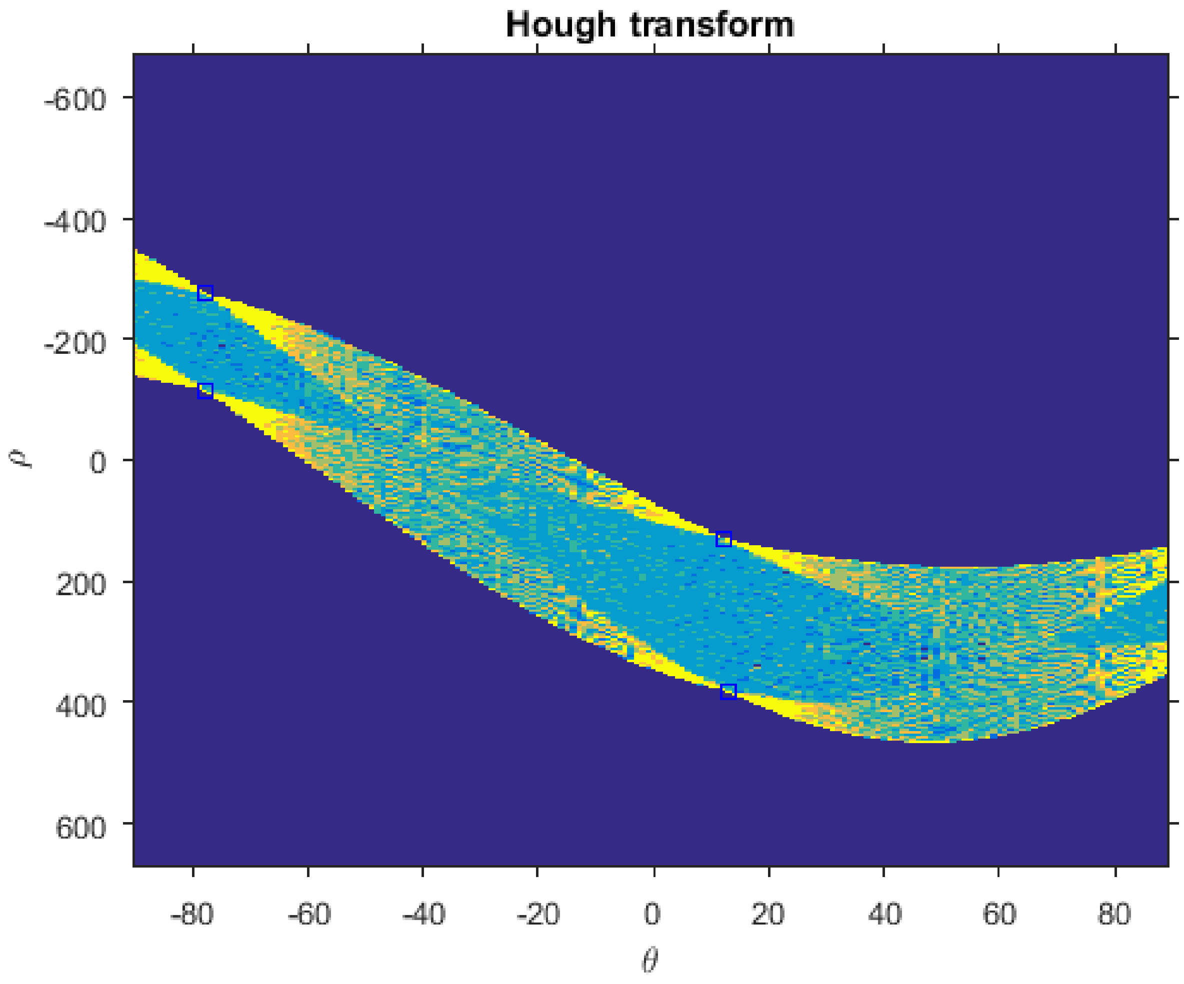
Example

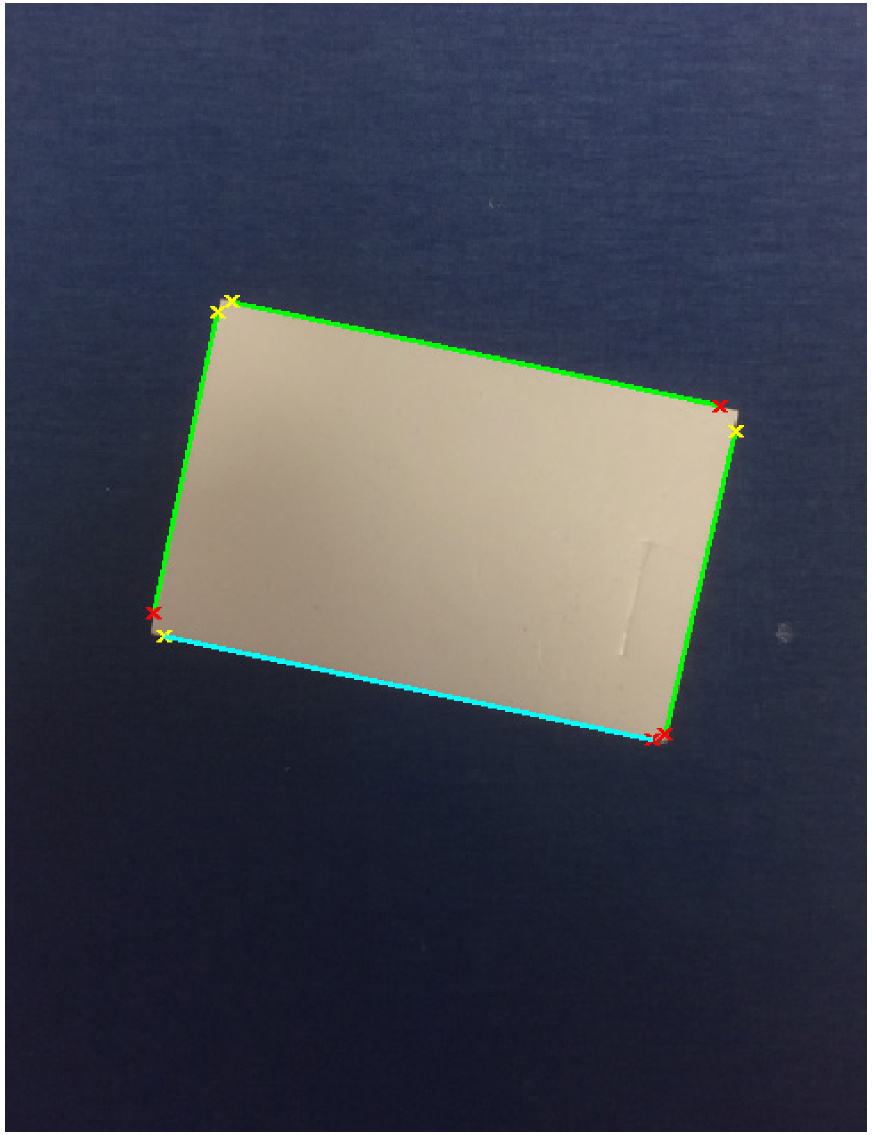
Original



Edge image (Canny)

Example (2)





Detected lines

Feature detection
Local keypoint features

- Corner detectors
  - Stable in space
  - Min eigenvalue, Harris

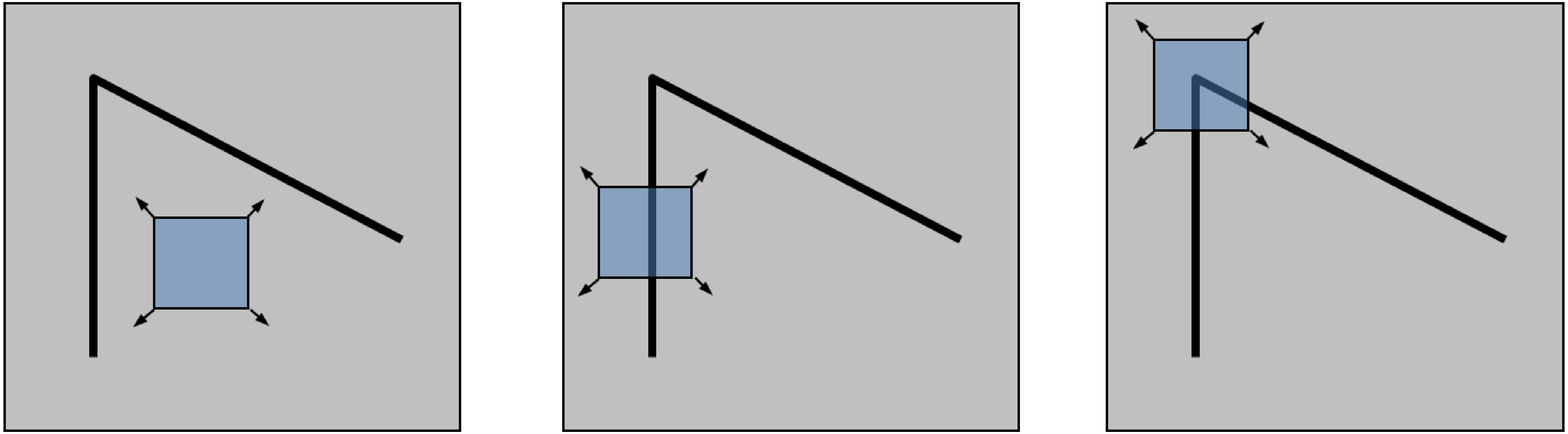
- Blob detectors
  - Stable in scale and space
  - LoG, DoG

Characteristics of good features

- Repeatability
- Distinctiveness
- Efficiency
- Locality

Local measure of feature distinctiveness

- Consider a small window of pixels around a feature
- How does the window change when you shift it?



"Flat" region:
No change in all directions

"Edge":
No change along edge

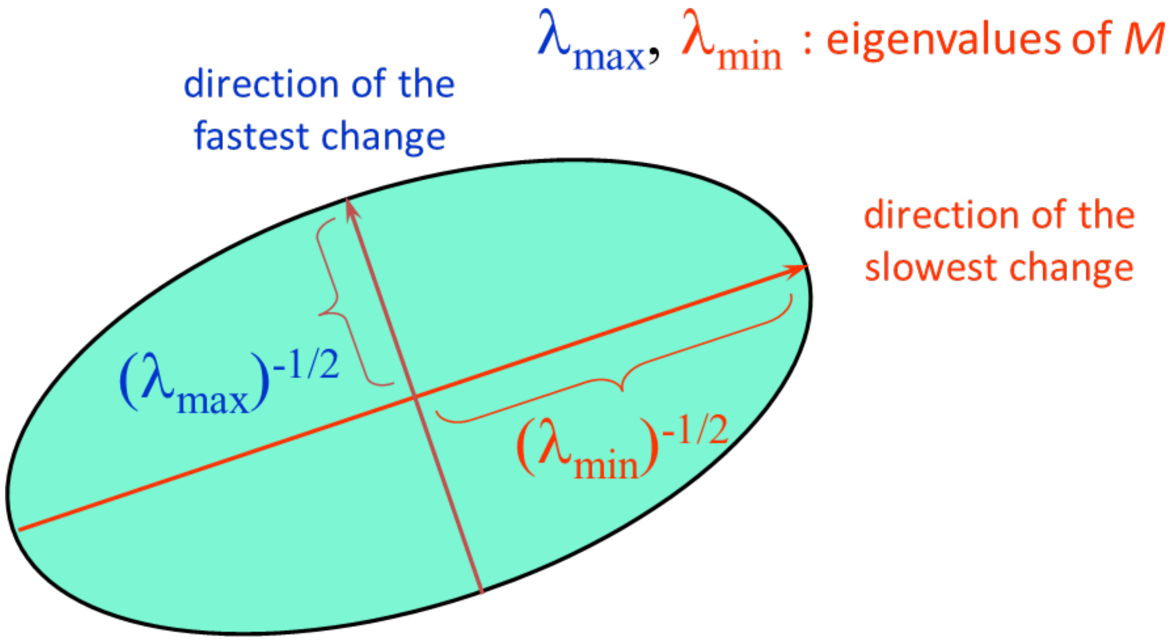
"Corner":
Change in all directions

Simplifying the measure even further

- Consider a horizontal "slice" of $E(u,v)$:

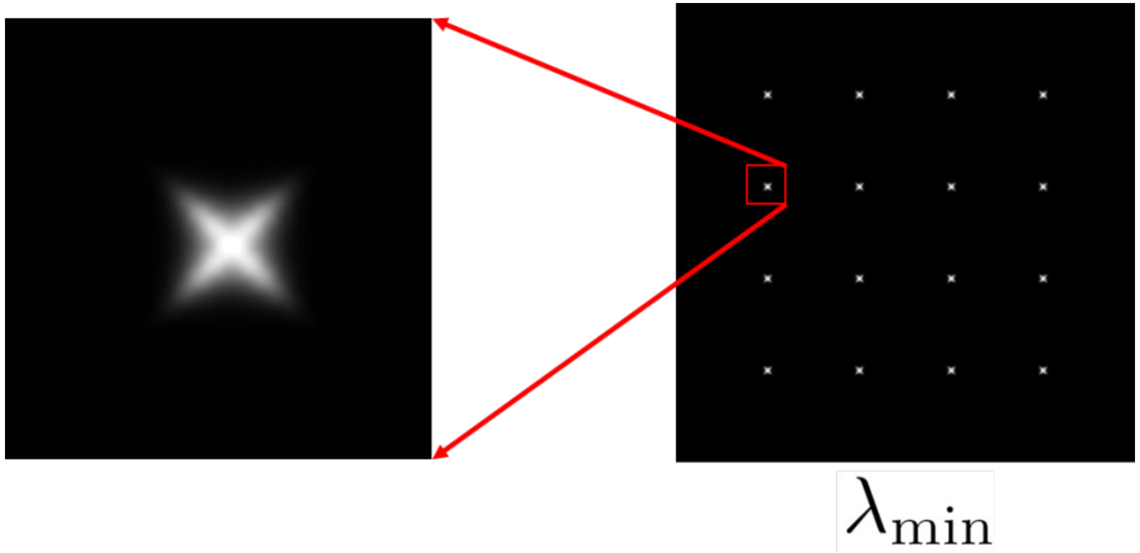
$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} = const$$

- This is the equation of an ellipse
  - Describe the surface using the eigenvalues of M



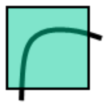
Corner detection summary

- Compute the gradient at each point in the image using derivatives of Gaussians
- Create the second moment matrix M from the entries in the gradient
- Compute the eigenvalues
- Find points with large response (λ_{min} > threshold)
- Choose those points where λ_{min} is a local maximum as features

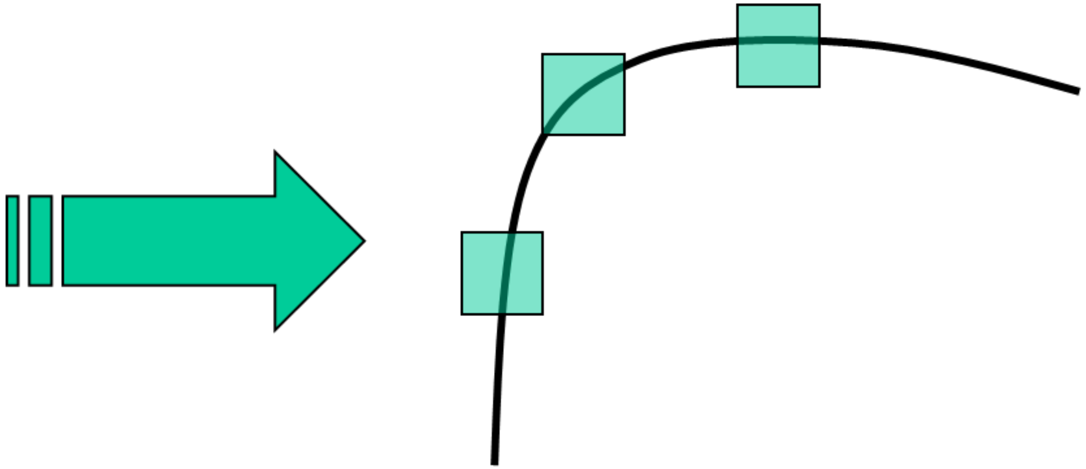


Harris detector properties

- **Scaling**



Corner



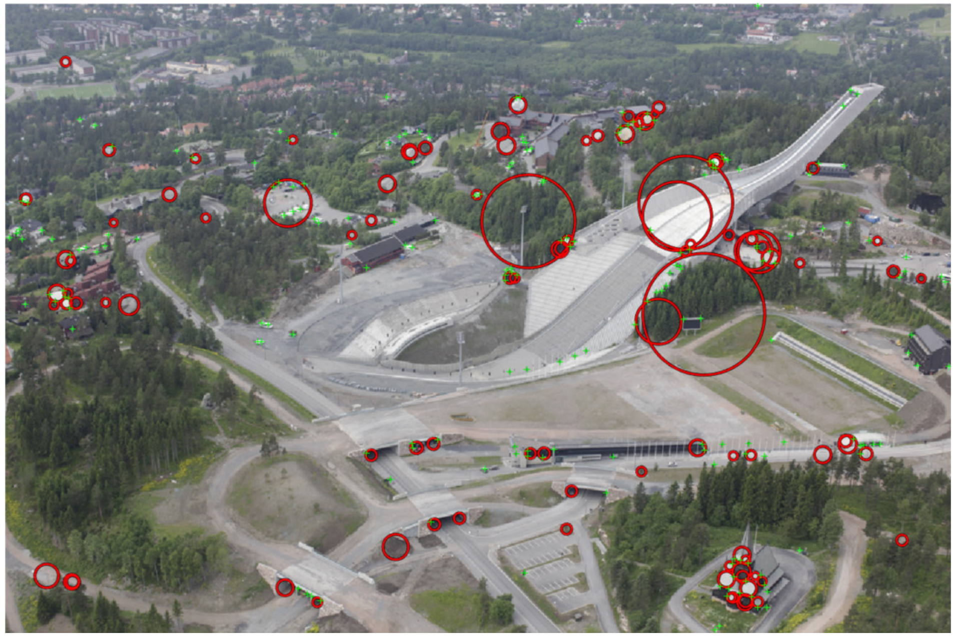
All points will be classified as edges

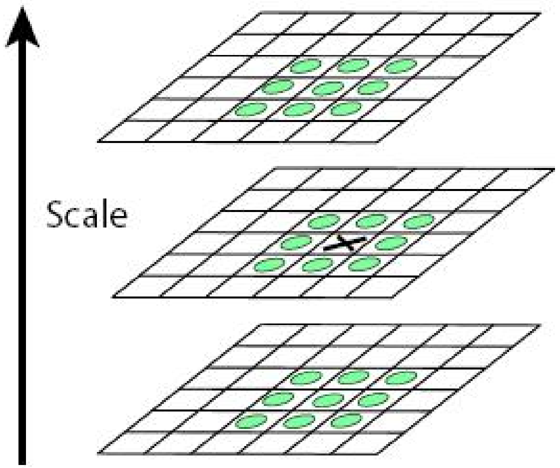
Corner location is not covariant to scaling!

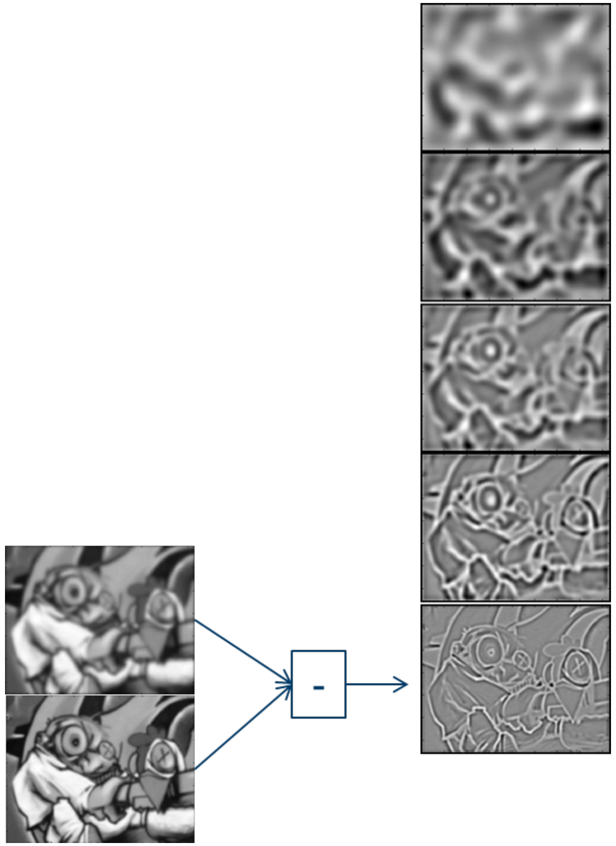
LoG blob detector

- Convolve the image with scale-normalized LoG at several scales
- Find maxima of squared LoG response in scale-space

- Approximate with Difference of Gaussians (DoG)

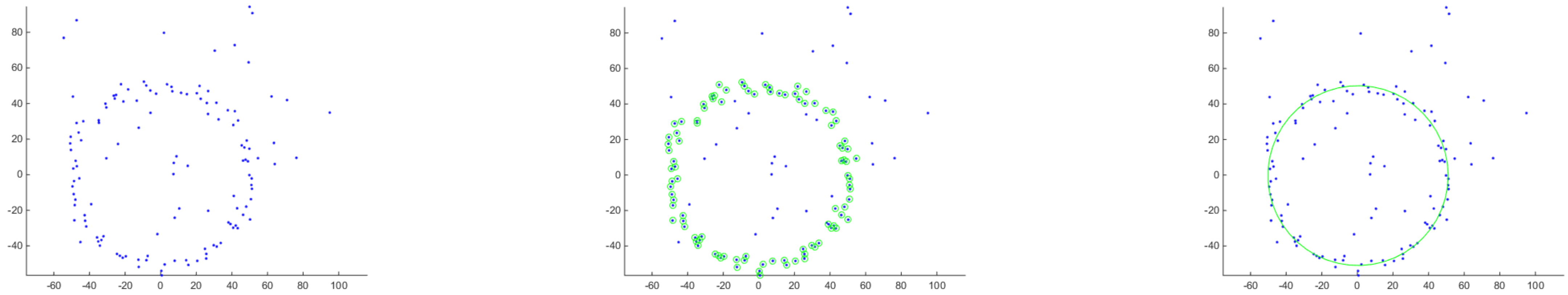






Feature detection
Robust estimation with RANSAC

- RANSAC
  - A robust iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers
  - Separates the observed data into “inliers” and “outliers”
  - Very useful if we want to use better, but less robust, estimation methods



RANSAC

Objective

Robustly fit a model $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha})$ to a data set $S = \{\boldsymbol{x}_i\}$

Algorithm

1. Determine a test model $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha}_{tst})$ from n random data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$

2. Check how well each individual data point in S fits with the test model
   - Data points within a distance t of the model constitute a set of inliers $S_{tst} \subseteq S$
   - Data points outside a distance t of the model are outliers

3. If S_{tst} is the largest set of inliers encountered so far, we keep this model
   - Set $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{tst}$ and $S_{IN} = S_{tst}$

4. Repeat steps 1-3 until N models have been tested

RANSAC

Comments

- Number of iterations required to achieve confidence p when testing random models from n-tuples of data elements from a dataset with inlier probability ω

$$N = \frac{log(1-p)}{log(1-\omega^n)}$$

- Typical desired level of confidence

$$p = 0.99$$

- Inlier probability ω is typically unknown, but can be estimated per iteration

$$\omega = \frac{\#max\ estimated\ inliers}{\#data\ elements}$$

- Instead of operating with a fixed and larger than necessary N we can update N for each iteration
  - **Adaptive RANSAC!**

		ω				
	N	0.9	0.8	0.7	0.6	0.5
	2	3	5	7	11	17
	3	4	7	11	19	35
	4	5	9	17	34	72
n	5	6	12	26	57	146
	6	7	16	37	97	293
	7	8	20	54	163	588
	8	9	26	78	272	1177

Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

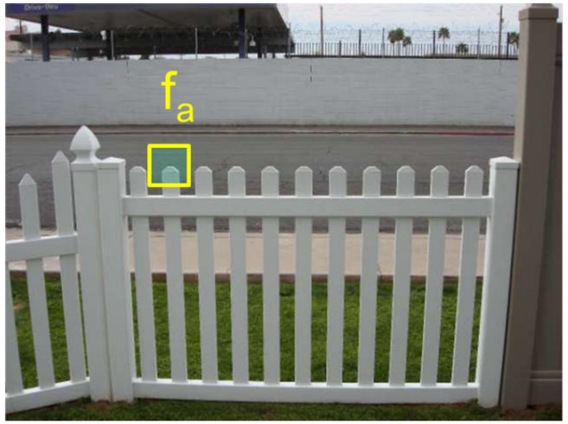
Feature matching
Feature descriptors and matching

- Matching keypoints
  - Comparing local patches in canonical scale and orientation

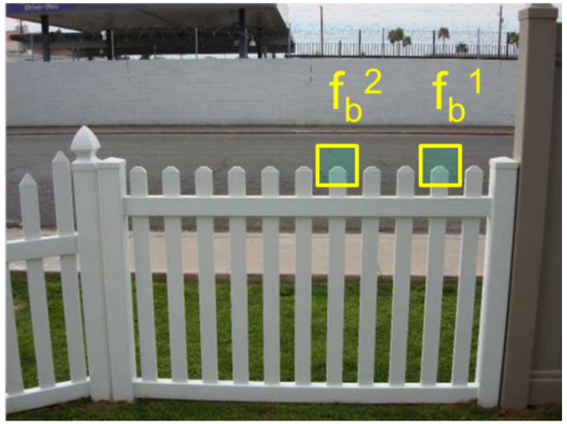
- Feature descriptors
  - Robust, distinctive and efficient

- Descriptor types
  - HoG descriptors
  - Binary descriptors

- Putative matching
  - Closest match, distance ratio, cross check



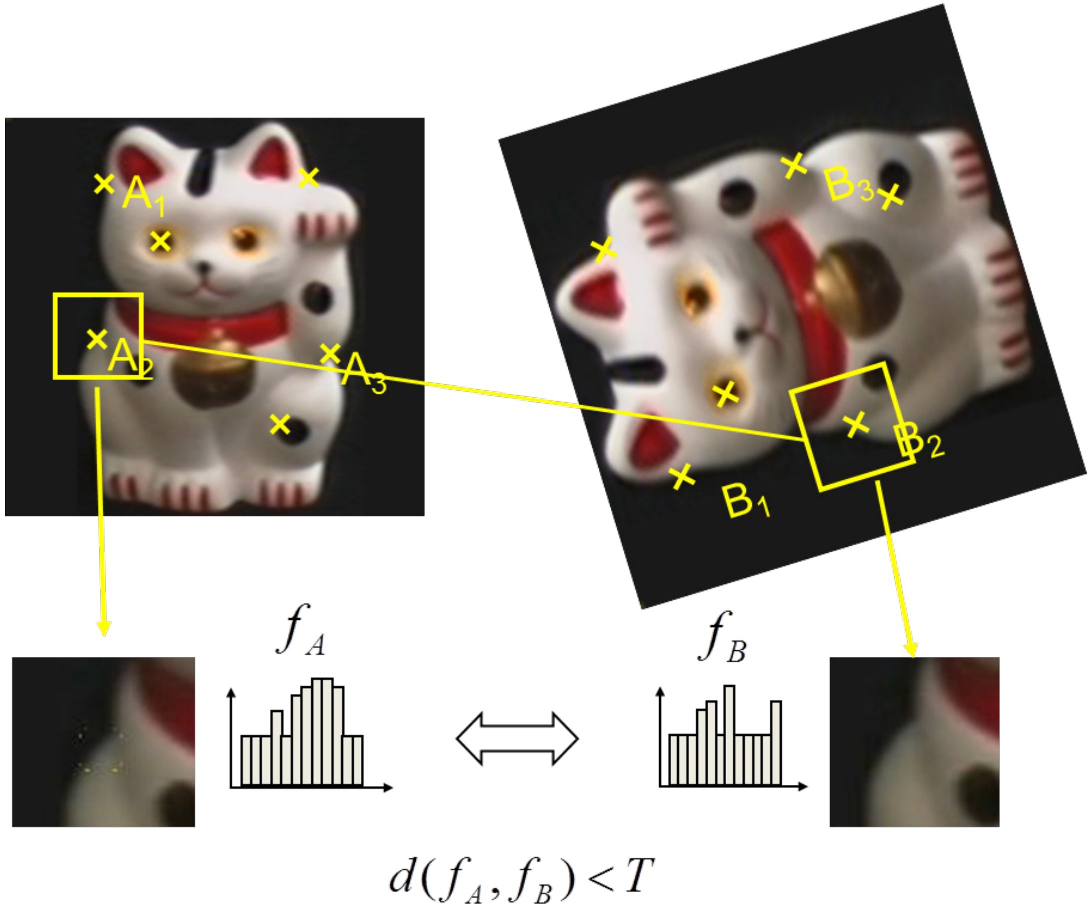
I_a



I_b

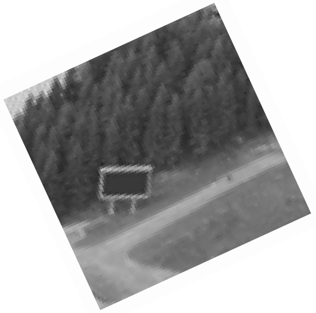
Feature matching

From keypoints to feature correspondences



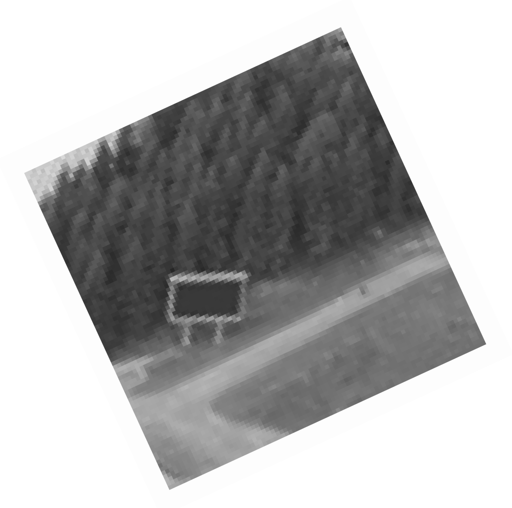
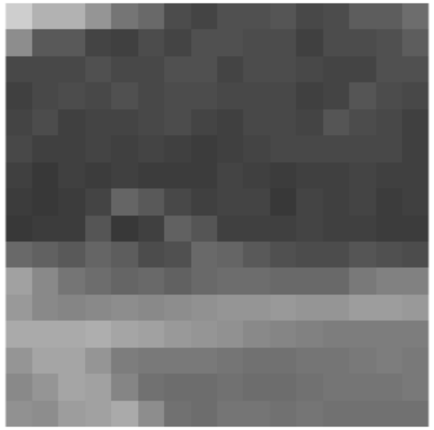
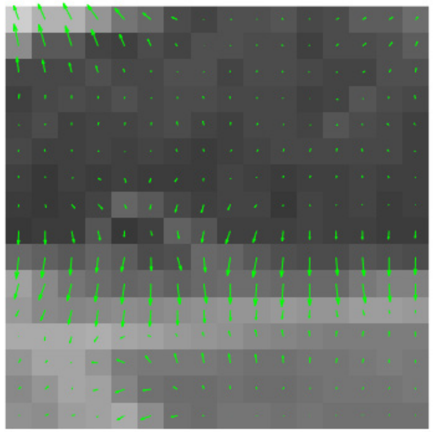
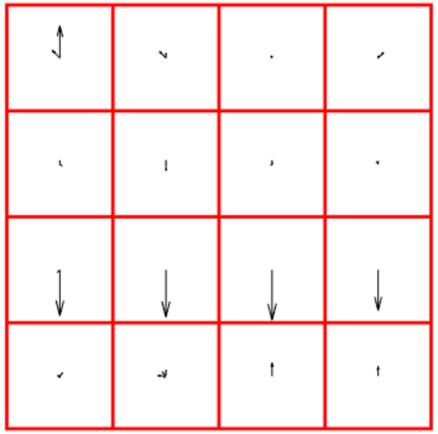
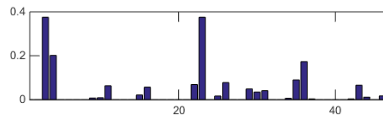
1. Detect a set of distinct feature points
2. Define a patch around each point
3. Extract and normalize the patch
4. Compute a local descriptor
5. Match local descriptors

Patch at detected position, scale, orientation



SIFT descriptor

- Extract a 16x16 patch around detected keypoint
- Compute the gradients and apply a Gaussian weighting function
- Divide the window into a 4x4 grid of cells
- Compute gradient direction histograms over 8 directions in each cell
- Concatenate the histograms to obtain a 128 dimensional feature vector
- Normalize to unit length

Binary descriptors

- Extremely efficient construction and comparison

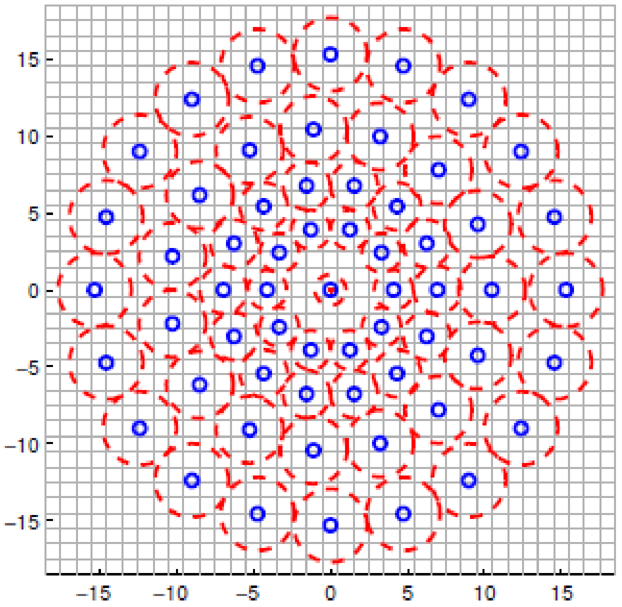
- Based on pairwise intensity comparisons
  - Sampling pattern around keypoint
  - Set of sampling pairs
  - Feature descriptor vector is a binary string:

$$F = \sum_{0 \le a \le N} 2^a T(P_a)$$

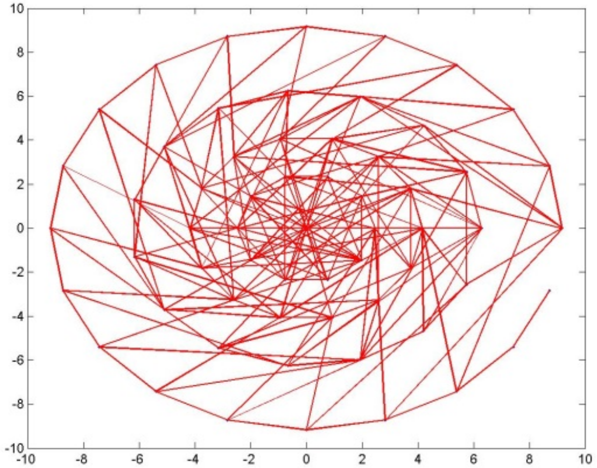
$$T(P_a) = \begin{cases} 1 & \text{if } I(P_a^{r1}) > I(P_a^{r2}) \\ 0 & \text{otherwise} \end{cases}$$

- Matching using Hamming distance:

$$L = \sum_{0 \le a \le N} XOR(F_a^1, F_a^2)$$



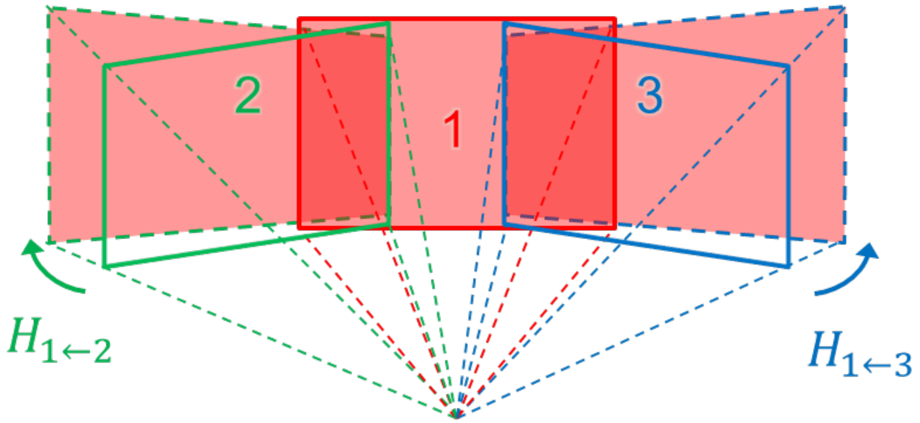
BRISK sampling pattern

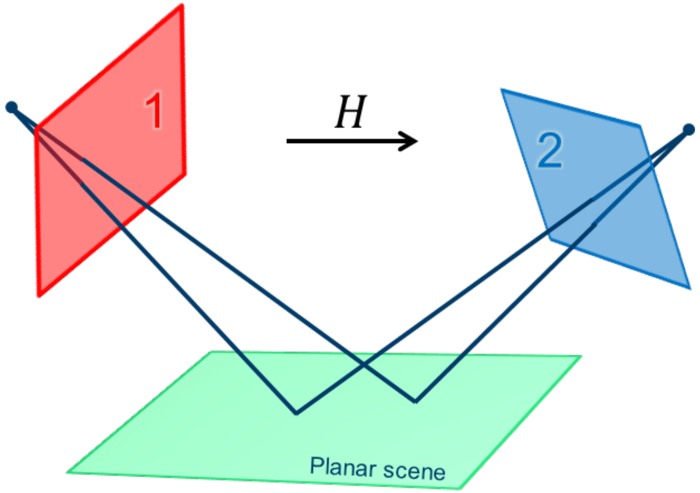


BRISK sampling pairs

Estimating homographies from feature correspondences

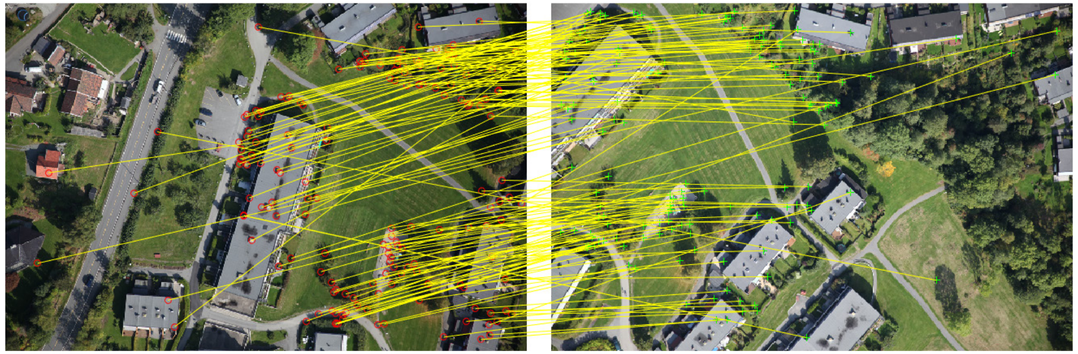
- Perspective images are sometimes perfectly related by a homography
  - Rotating camera
  - Planar scene
- Point-correspondences $\tilde{\mathbf{u}}_i \leftrightarrow \tilde{\mathbf{u}}'_i$ can be established automatically between two such images
  - Wrong correspondences are common
- The homography can be estimated from the point correspondences
  - Need at least 4
  - Robust estimation techniques are recommended





Estimating homographies from feature correspondences

- RANSAC estimation of homography $\mathbf{H}\tilde{\mathbf{u}} = \tilde{\mathbf{u}}'$
  - Direct Linear Transform (DLT) on 4 random correspondences $\tilde{\mathbf{u}}_i \leftrightarrow \tilde{\mathbf{u}}_i'$
  - Inliers have a small reprojection error
    $\epsilon_i = d(\mathbf{H}\mathbf{u}_i, \mathbf{u'}_i) + d(\mathbf{u}_i, \mathbf{H}^{-1}\mathbf{u'}_i)$



- The RANSAC estimated homography is random
  - Only estimated from 4 correspondences!

- A “better” homography can be estimated based on all the inlier correspondences
  - Normalized DLT
  - Iterative methods

- Using the homography we can warp one image into the coordinate frame of the other

Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

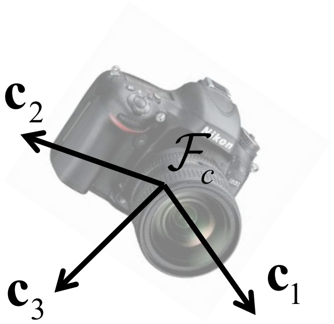
WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

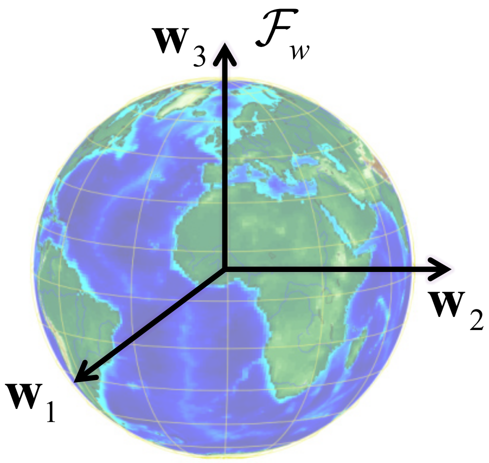
- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Orientation – Several representations



- Orientation of a frame $\mathcal{F}_b$ relative to a frame $\mathcal{F}_a$ has several representations

Representation	
– Rotation matrix $\mathbf{R} \in SO(3)$	Main representation for us!
– Euler angles $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^T$	Minimal representation
– Axis-angle $(\mathbf{v}, \phi) = \{[v_1, v_2, v_3]^T, \phi\}$	We will not us this
– Unit quaternion $\mathbf{q} = q_1 + q_2 i + q_3 j + q_4 k$	We will use this indirectly



- Important properties

Property	
– Inverse	$\mathbf{R}_{ba} = \mathbf{R}_{ab}^{-1}$
– Composition	$\mathbf{R}_{ac} = \mathbf{R}_{ab}\mathbf{R}_{bc}$
– Action on points	$\mathbf{x}^b = \mathbf{R}_{ba}\mathbf{x}^a$

$$\mathbf{R}_{wc} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \begin{matrix} \mathbf{w}_1^{cT} \\ \mathbf{w}_2^{cT} \\ \mathbf{w}_3^{cT} \end{matrix}$$

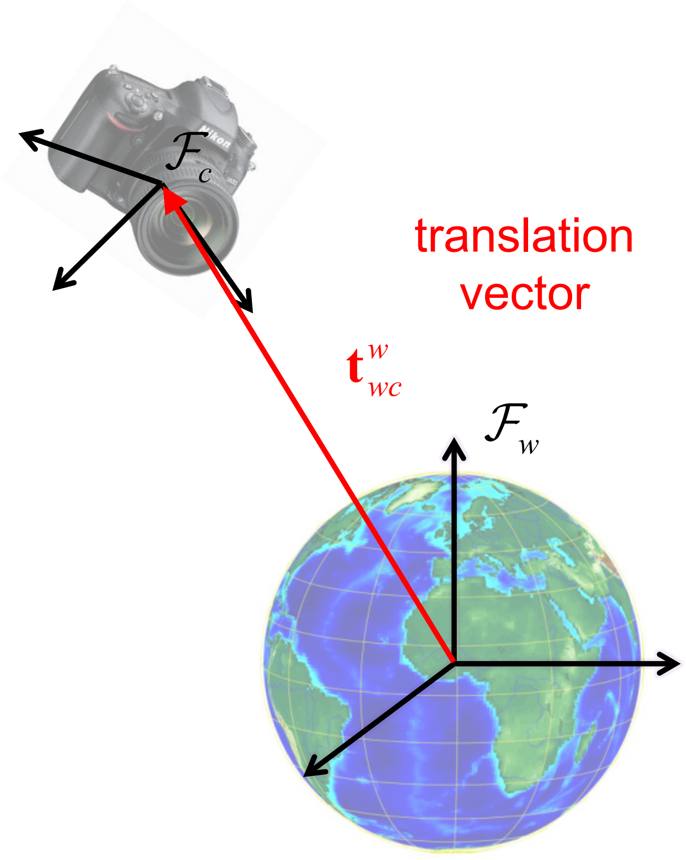
$$\mathbf{c}_1^w \quad \mathbf{c}_2^w \quad \mathbf{c}_3^w$$

Pose

- The pose of the camera frame $\mathcal{F}_c$ with respect to the world frame $\mathcal{F}_w$ can be represented by the Euclidean transformation matrix

$$\mathbf{T}_{wc} = \begin{bmatrix} \mathbf{R}_{wc} & \mathbf{t}_{wc}^{w} \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3)$$

where $\mathbf{R}_{wc} \in SO(3)$ is a rotation matrix and $\mathbf{t}_{wc}^{w} \in \mathbb{R}^3$ is a translation vector given in world coordinates



NOTATION

$\mathbf{T}_{ab}$ = The pose of $\mathcal{F}_b$ relative to $\mathcal{F}_a$

$\mathbf{R}_{ab}$ = The orientation of $\mathcal{F}_b$ relative to $\mathcal{F}_a$

$\mathbf{t}_{ab}^{c}$ = The translation of $\mathcal{F}_b$ relative to $\mathcal{F}_a$ given in $\mathcal{F}_c$ coordinates

Pose – Inverse

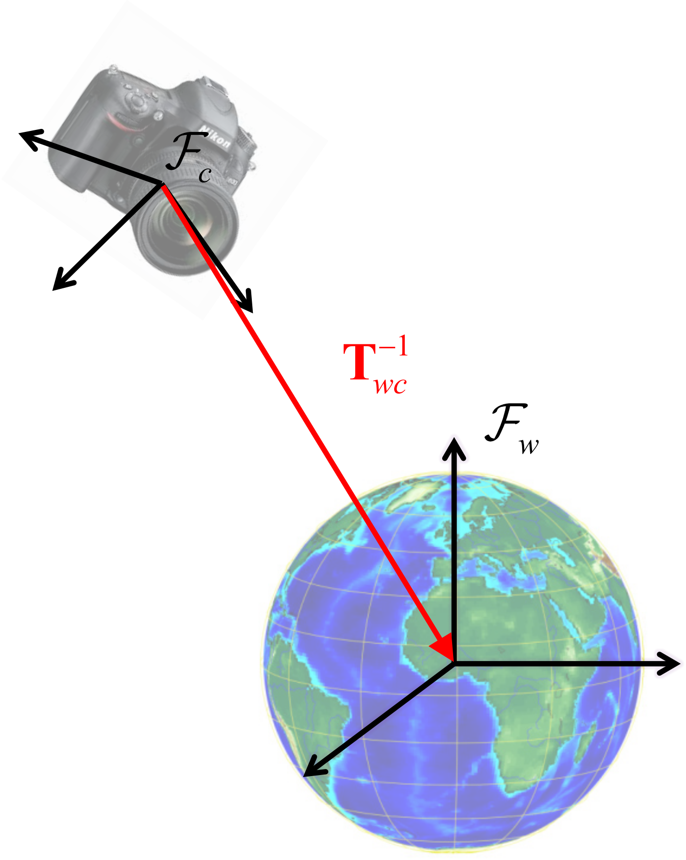
- The opposite pose, the pose of $\mathcal{F}_w$ with respect to $\mathcal{F}_c$, is given by the inverse transformation

$$\mathbf{T}_{cw} = \mathbf{T}_{wc}^{-1}$$

- One can show that

$$\mathbf{T}_{cw} = \begin{bmatrix} \mathbf{R}_{wc} & \mathbf{t}_{wc}^{w} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}_{wc}^{T} & -\mathbf{R}_{wc}^{T}\mathbf{t}_{wc}^{w} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

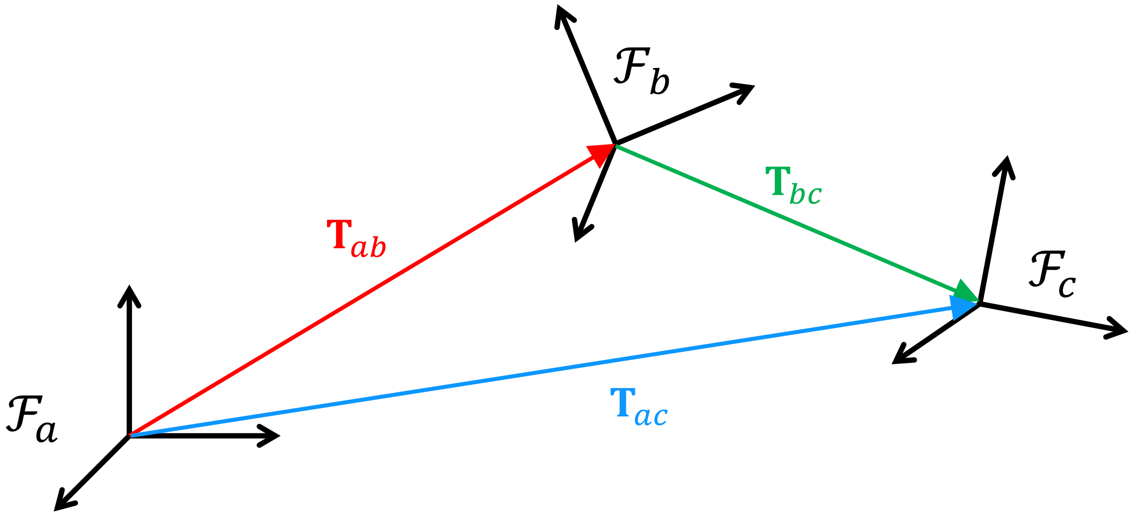
- Hence $\mathbf{R}_{cw} = \mathbf{R}_{wc}^{T}$ and $\mathbf{t}_{cw}^{c} = -\mathbf{R}_{wc}^{T}\mathbf{t}_{wc}^{w}$



Pose – Composition

We can chain together consecutive poses by compounding transformation matrices

$$\mathbf{T}_{ac} = \mathbf{T}_{ab}\mathbf{T}_{bc}$$



Note
The indexes are always pairwise equal

$$\tilde{\mathbf{x}}^{a} = \mathbf{T}_{ab}\mathbf{T}_{bc}\tilde{\mathbf{x}}^{c}$$

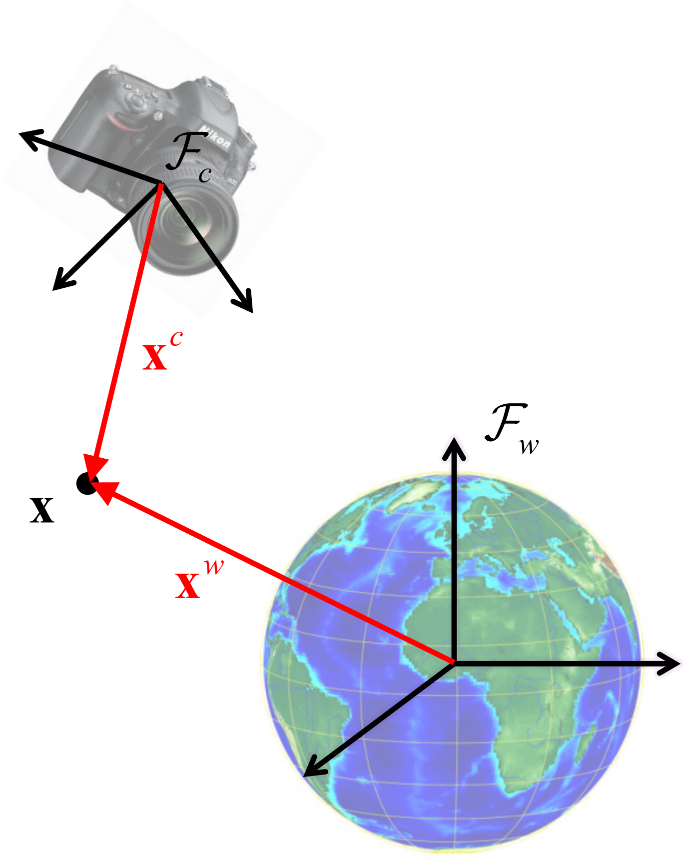
destination frame (a) — intermediate frame (b) — source frame (c)

Pose – Action on points

- The matrix $\mathbf{T}_{cw}$ represents the pose of $\mathcal{F}_w$ relative to $\mathcal{F}_c$, but it is also a point transformation from $\mathcal{F}_w$ to $\mathcal{F}_c$
- A point $\mathbf{x}^w$ in world coordinates can be transformed to camera coordinates by

$$\tilde{\mathbf{x}}^c = \mathbf{T}_{cw}\tilde{\mathbf{x}}^w$$

$$\mathbf{x}^c = \mathbf{R}_{cw}\mathbf{x}^w + \mathbf{t}^c_{cw}$$



Note

The indexes are always pairwise equal

$$\tilde{\mathbf{x}}^{a} = \mathbf{T}_{ab}\tilde{\mathbf{x}}^{b}$$

destination frame

source frame

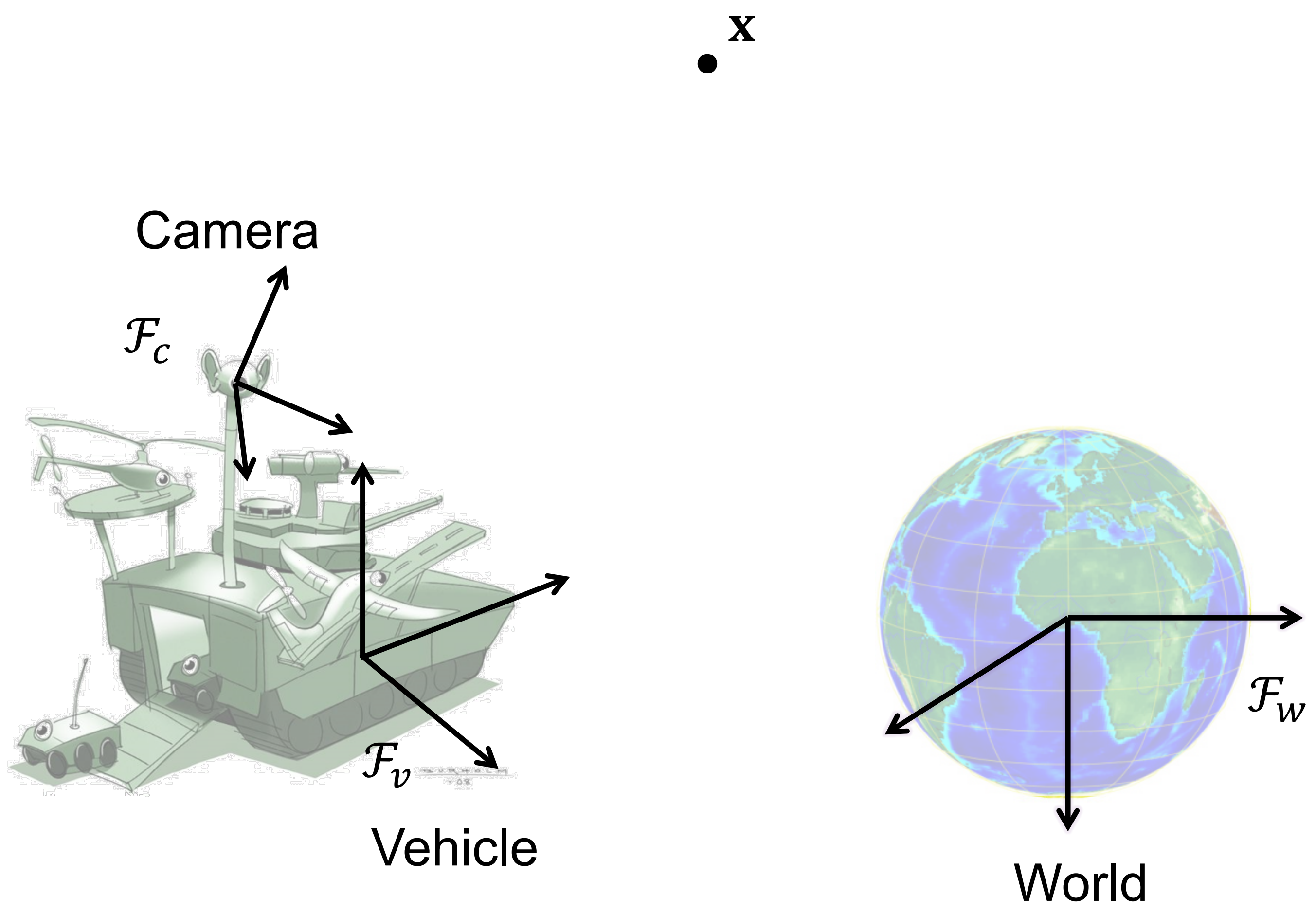
Example – Camera on a vehicle in the world

A point $\mathbf{x}$ has a known position relative to a camera mounted on a vehicle

The vehicle has a known pose relative to the world

The camera has a known pose relative to the vehicle

Find expressions for $\mathbf{x}^v$ and $\mathbf{x}^w$



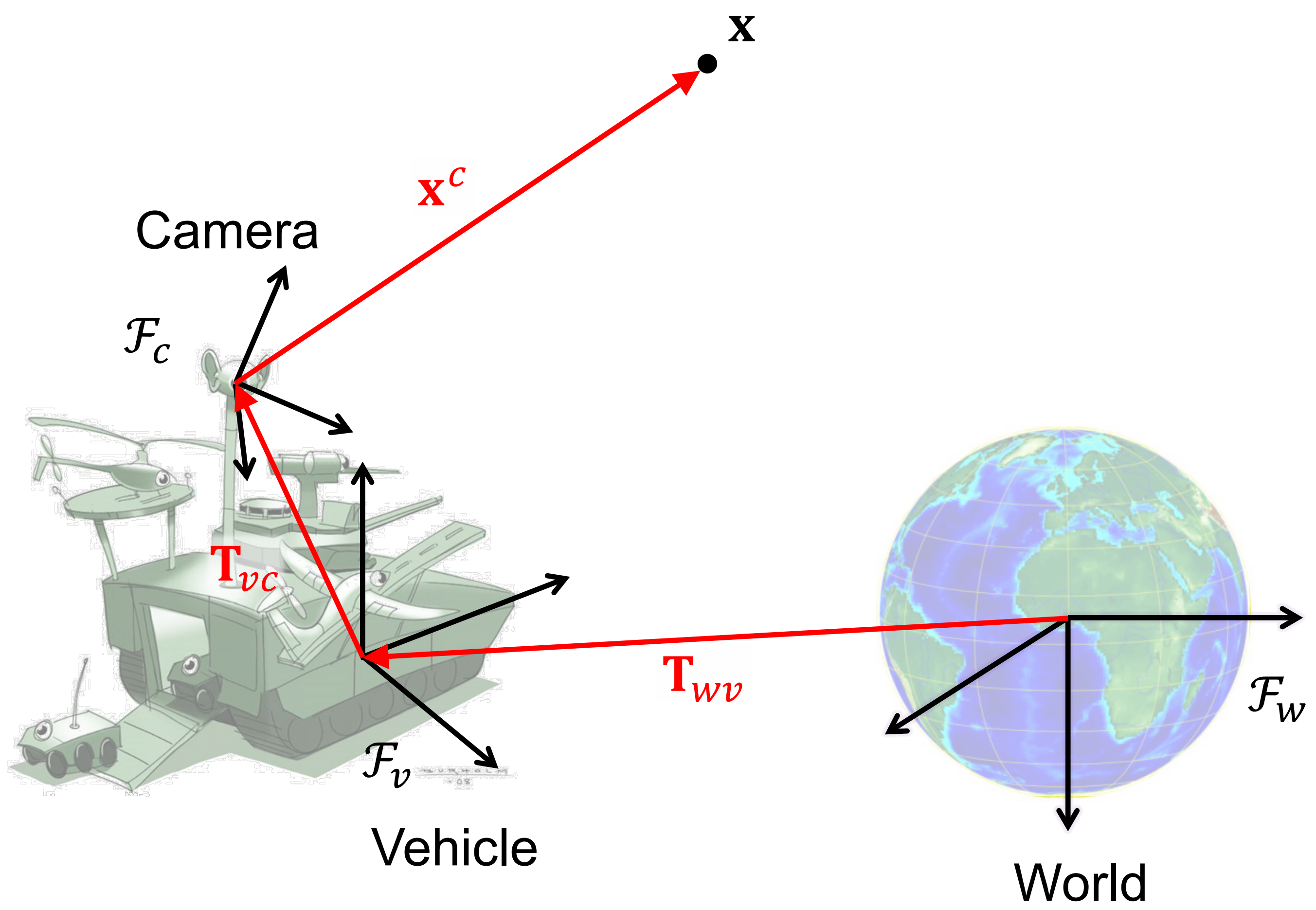
Example – Camera on a vehicle in the world

A point $\mathbf{x}$ has a known position relative to a camera mounted on a vehicle $\mathbf{x}^c$

The vehicle has a known pose relative to the world $\mathbf{T}_{wv}$

The camera has a known pose relative to the vehicle $\mathbf{T}_{vc}$

Find expressions for $\mathbf{x}^v$ and $\mathbf{x}^w$



Example – Camera on a vehicle in the world

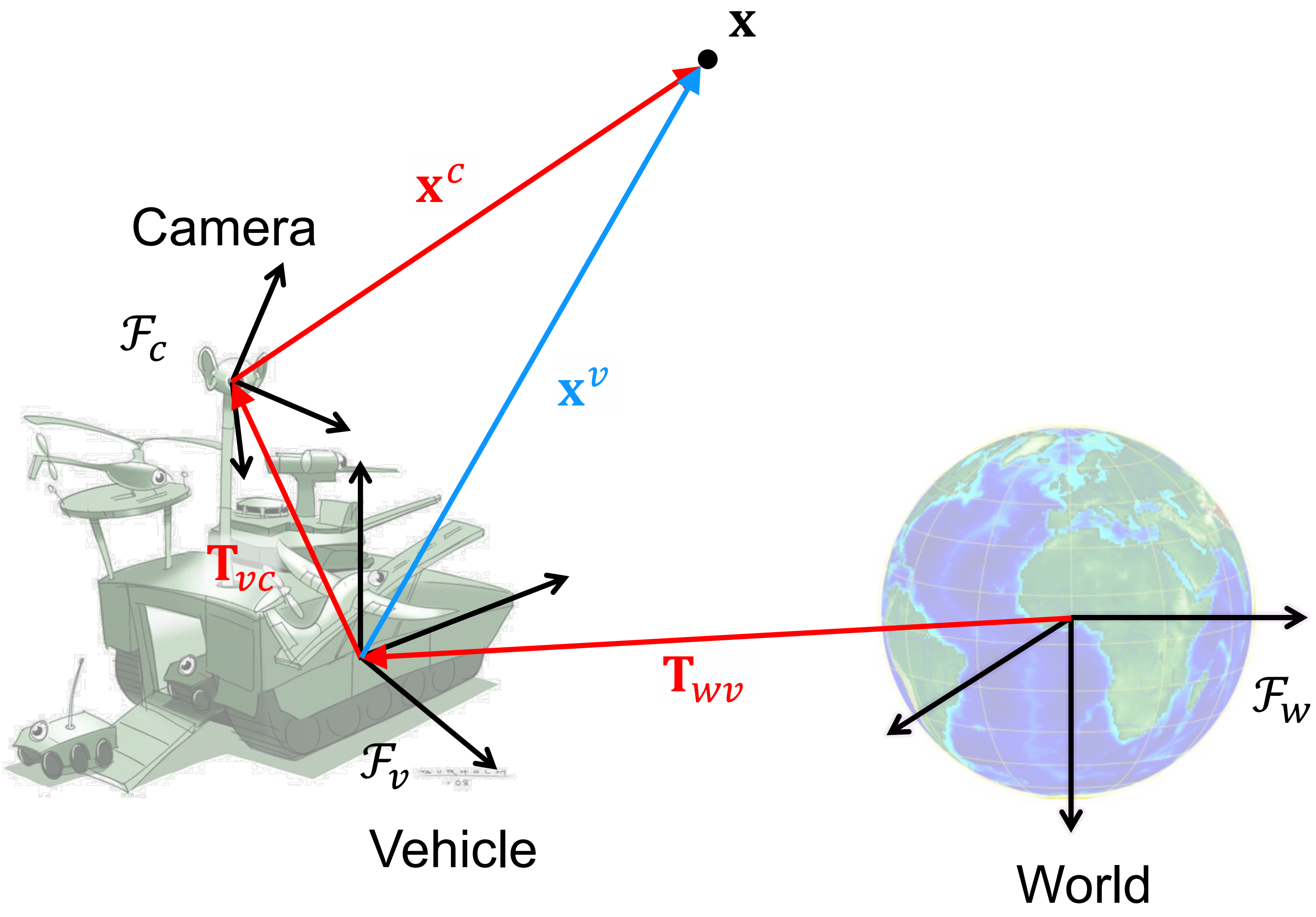
A point $\mathbf{x}$ has a known position relative to a camera mounted on a vehicle $\mathbf{x}^c$

The vehicle has a known pose relative to the world $\mathbf{T}_{wv}$

The camera has a known pose relative to the vehicle $\mathbf{T}_{vc}$

Find expressions for $\mathbf{x}^v$ and $\mathbf{x}^w$

$$\tilde{\mathbf{x}}^v = \mathbf{T}_{vc}\tilde{\mathbf{x}}^c$$



Example – Camera on a vehicle in the world

A point $\mathbf{x}$ has a known position relative to a camera mounted on a vehicle $\mathbf{x}^c$

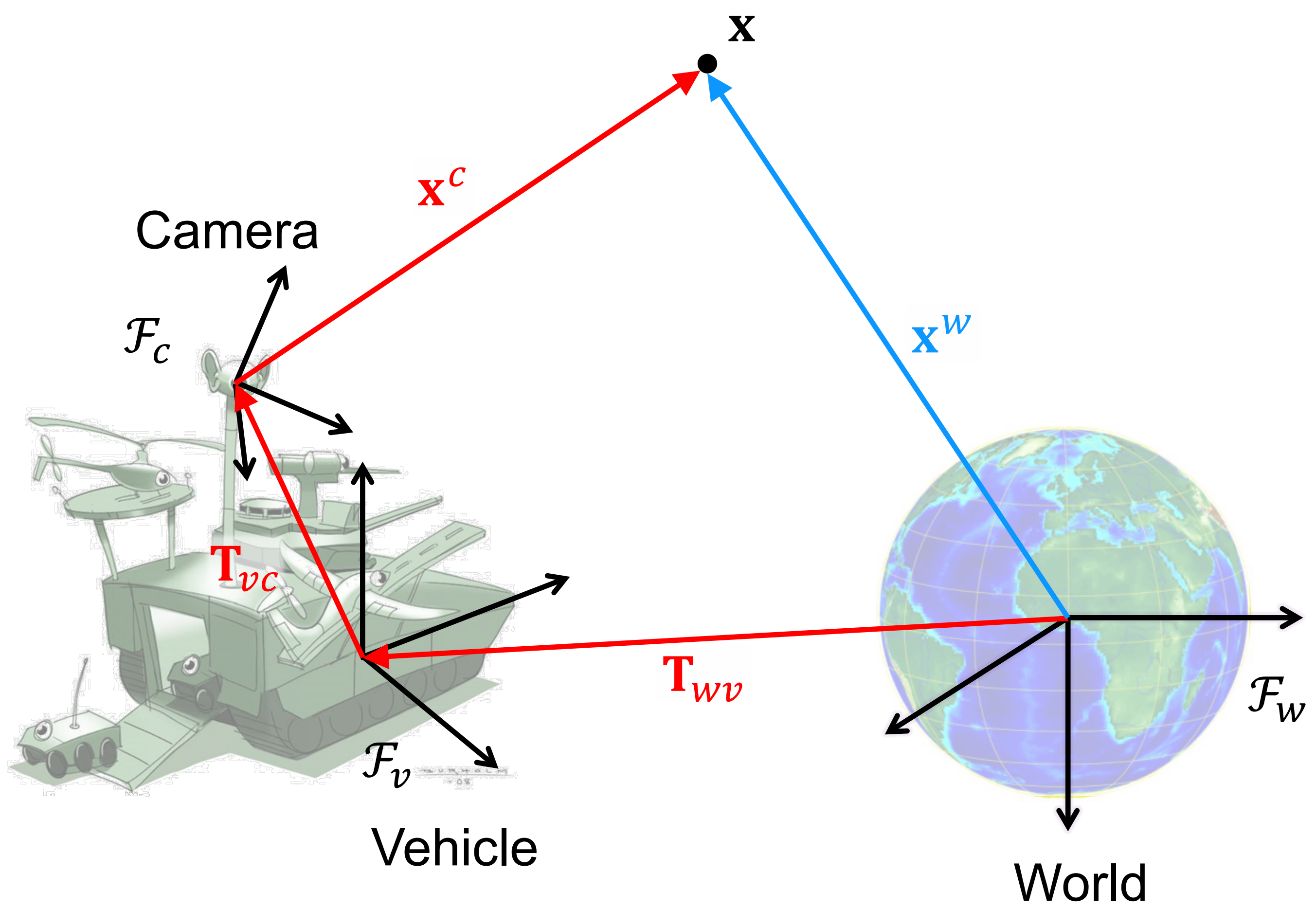
The vehicle has a known pose relative to the world $\mathbf{T}_{wv}$

The camera has a known pose relative to the vehicle $\mathbf{T}_{vc}$

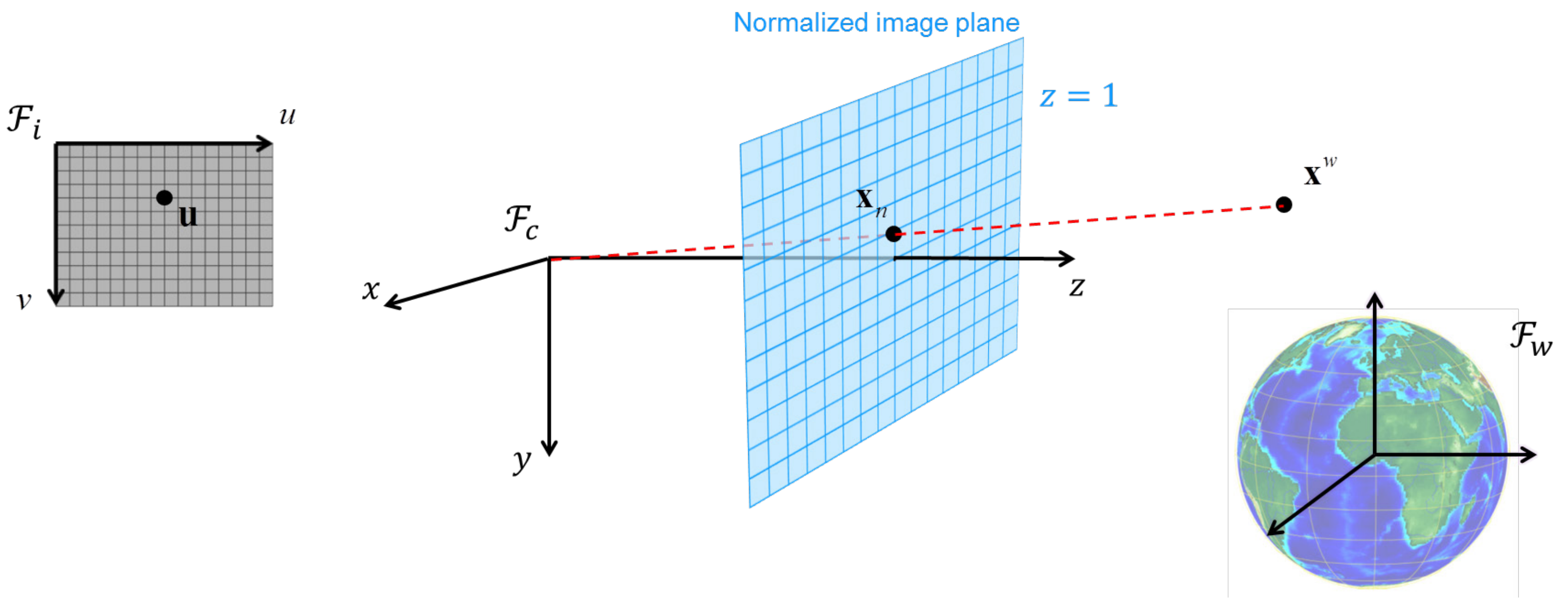
Find expressions for $\mathbf{x}^v$ and $\mathbf{x}^w$

$$\tilde{\mathbf{x}}^v = \mathbf{T}_{vc}\tilde{\mathbf{x}}^c$$

$$\tilde{\mathbf{x}}^w = \mathbf{T}_{wv}\mathbf{T}_{vc}\tilde{\mathbf{x}}^c$$



The perspective camera model revisited



Pose of $\mathcal{F}_w$ relative to $\mathcal{F}_c$!

- The perspective camera model when we consider 3D points in a frame $\mathcal{F}_w$ instead of the camera frame $\mathcal{F}_c$

$$\tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw}^{c} \end{bmatrix}\tilde{\mathbf{x}}^{w}$$

$$\tilde{\mathbf{u}} = \underbrace{\begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{\Pi}_0} \underbrace{\begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw}^{c} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{T}_{cw}} \tilde{\mathbf{x}}^{w}$$

Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

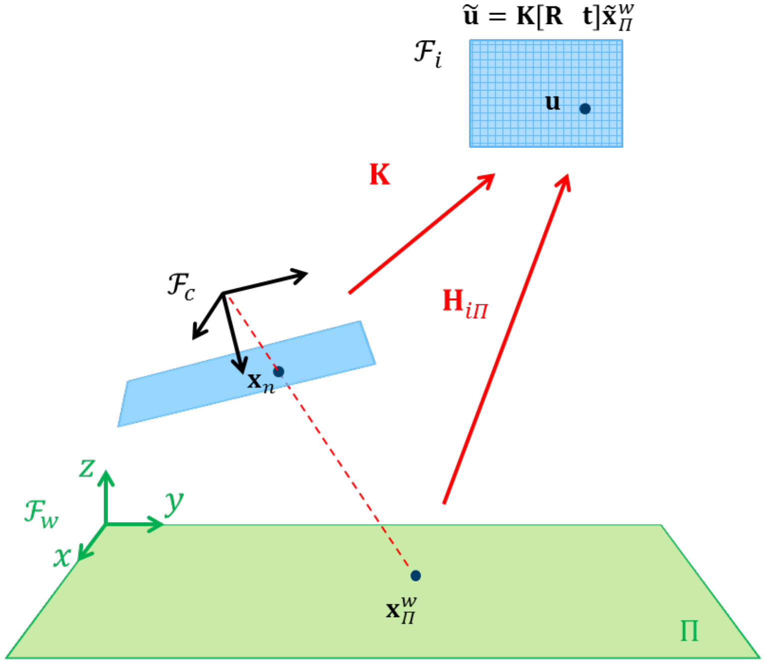
WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Pose from a known 3D map

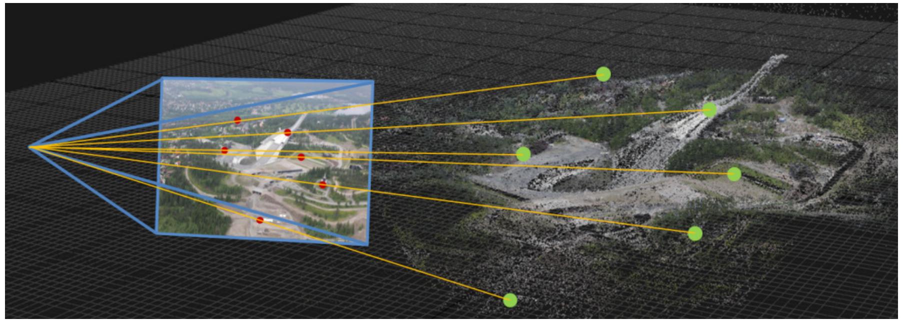


- Homography-based method

  For a calibrated camera,
  we have a relation between the camera pose
  and the homography between the world plane and the image!

$$\mathbf{H}_{i\Pi} = \mathbf{K}\left[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}\right] \qquad \mathbf{T}_{cw} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

- Indirect methods based on minimizing *geometric error*

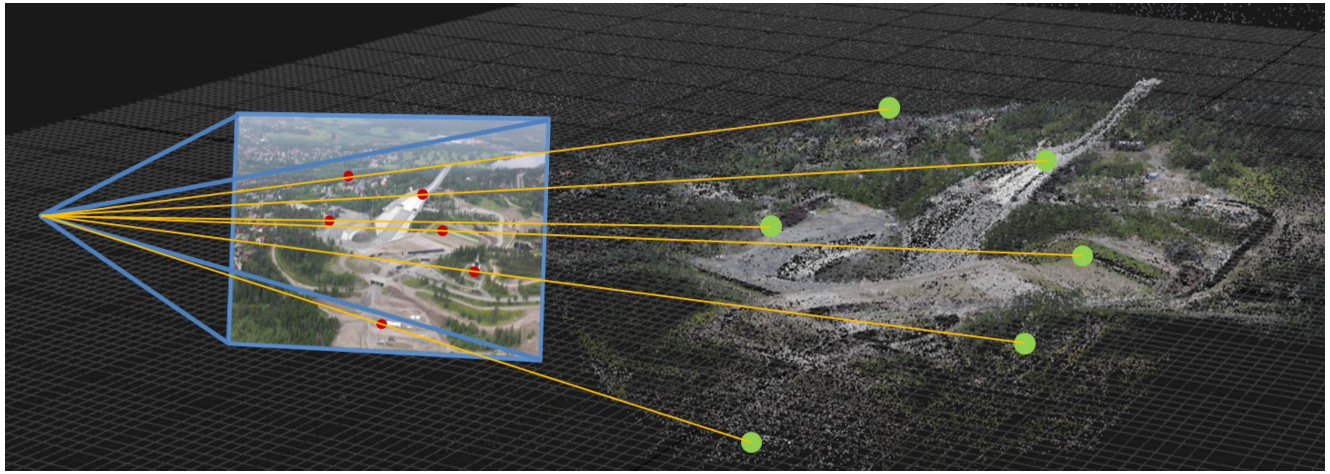


$$\mathbf{T}_{cw}^{*} = \underset{\mathbf{T}_{cw}}{\operatorname{argmin}} \sum_i \left\| \pi(\mathbf{T}_{cw}\tilde{\mathbf{x}}_i^{w}) - \mathbf{u}_i \right\|^2$$

How can we solve the indirect tracking problem?

Minimize **geometric error** with **nonlinear least squares!**

$$\mathbf{T}_{cw}^{*} = \underset{\mathbf{T}_{cw}}{\operatorname{argmin}} \sum_{i} \left\| \pi(\mathbf{T}_{cw} \tilde{\mathbf{x}}_{i}^{w}) - \mathbf{u}_{i} \right\|^{2}$$



Nonlinear least squares

We can find the MAP estimate
of our unknown states given measurements

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\ p(X \mid Z)$$

by representing it as
a nonlinear least squares problem

$$X^* = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2$$

Choose a suitable inital estimate X^0

$$\mathbf{A}, \mathbf{b} \leftarrow \text{Linearize at } X^t$$

$$\mathbf{\Delta}^* \leftarrow \text{Solve } \underset{\mathbf{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A}\mathbf{\Delta} - \mathbf{b} \right\|^2$$

$$X^{t+1} \leftarrow X^t + \mathbf{\Delta}^*$$

Nonlinear least squares

We can find the MAP estimate
of our unknown states given measurements

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\ p(X \mid Z)$$

by representing it as
a nonlinear least squares problem

$$X^{*} = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2$$

Choose a suitable inital estimate X^0

$\mathbf{A}, \mathbf{b} \leftarrow \text{Linearize at } X^t$

- Jacobians
- Covariance weighting

$\boldsymbol{\Delta}^* \leftarrow \text{Solve } \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\Delta} - \mathbf{b} \right\|^2$

$X^{t+1} \leftarrow X^t + \boldsymbol{\Delta}^*$

Example:
Range-based localization

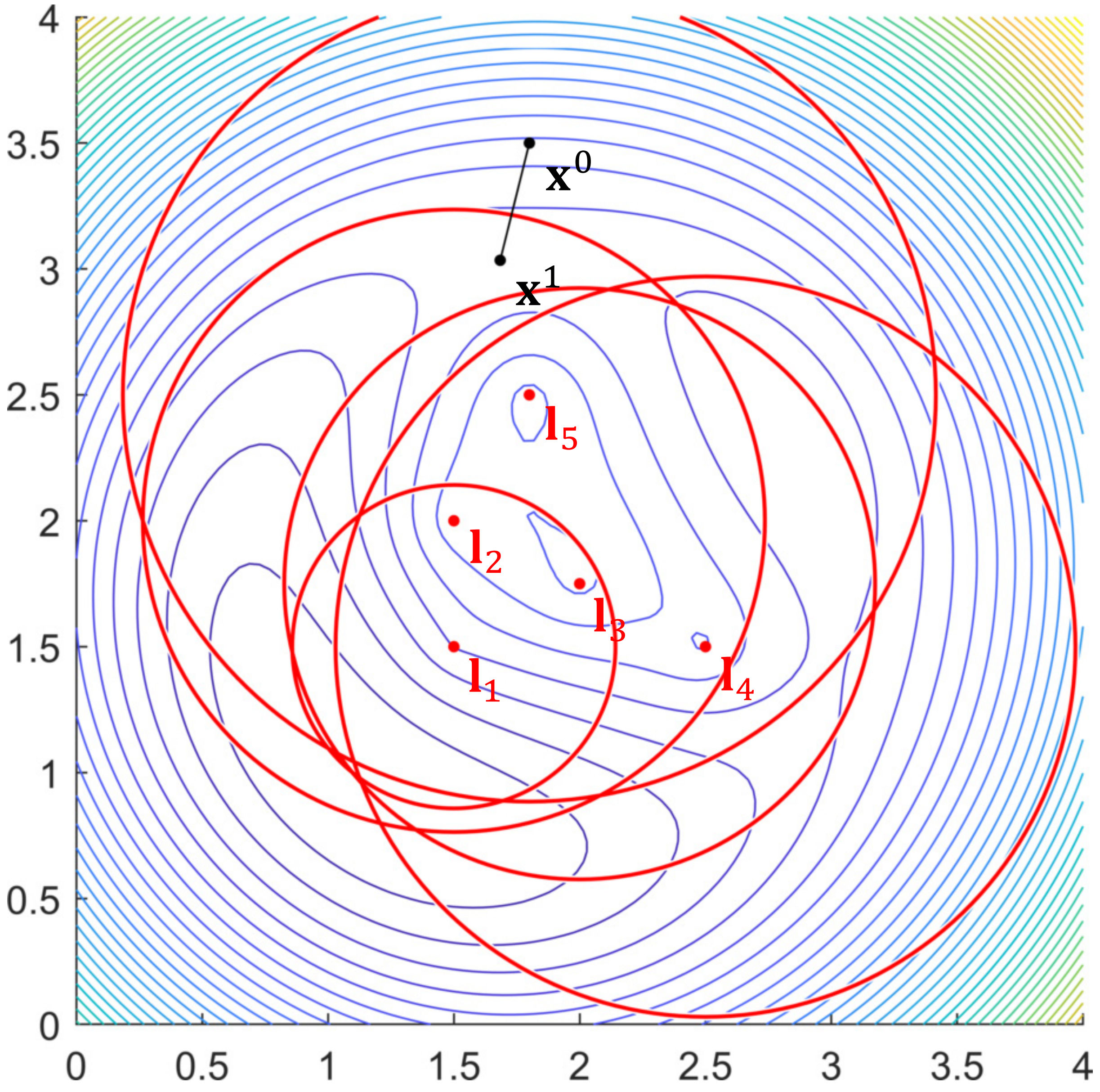
Linearized problem at $\mathbf{x}^0$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\delta} - \mathbf{b} \right\|^2$$

$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.99 \\ 0.20 & 0.98 \\ -0.11 & 0.99 \\ -0.33 & 0.94 \\ 0 & 1.00 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -1.38 \\ -0.29 \\ -0.59 \\ -0.65 \\ 0.62 \end{bmatrix}$$

Solution to the normal equations $\mathbf{A}^T\mathbf{A}\boldsymbol{\delta}^* = \mathbf{A}^T\mathbf{b}$:

$$\boldsymbol{\delta}^* = \begin{bmatrix} -0.12 \\ -0.47 \end{bmatrix} \qquad \mathbf{x}^1 = \mathbf{x}^0 + \boldsymbol{\delta}^* = \begin{bmatrix} 1.68 \\ 3.03 \end{bmatrix}$$



Nonlinear least squares

We can find the MAP estimate
of our unknown states given measurements

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\ p(X \mid Z)$$

by representing it as
a nonlinear least squares problem

$$X^{*} = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^{2}$$

Choose a suitable inital estimate X^0

$$\mathbf{A}, \mathbf{b} \leftarrow \text{Linearize at } X^t$$

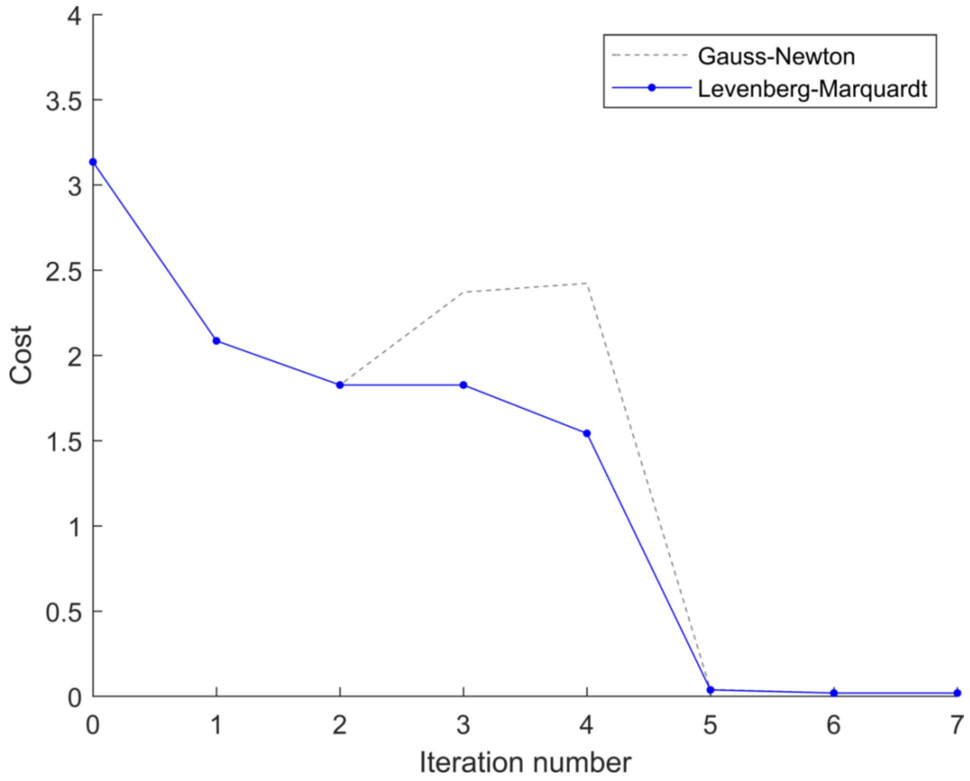
$$\boldsymbol{\Delta}^{*} \leftarrow \text{Solve } \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\Delta} - \mathbf{b} \right\|^{2}$$

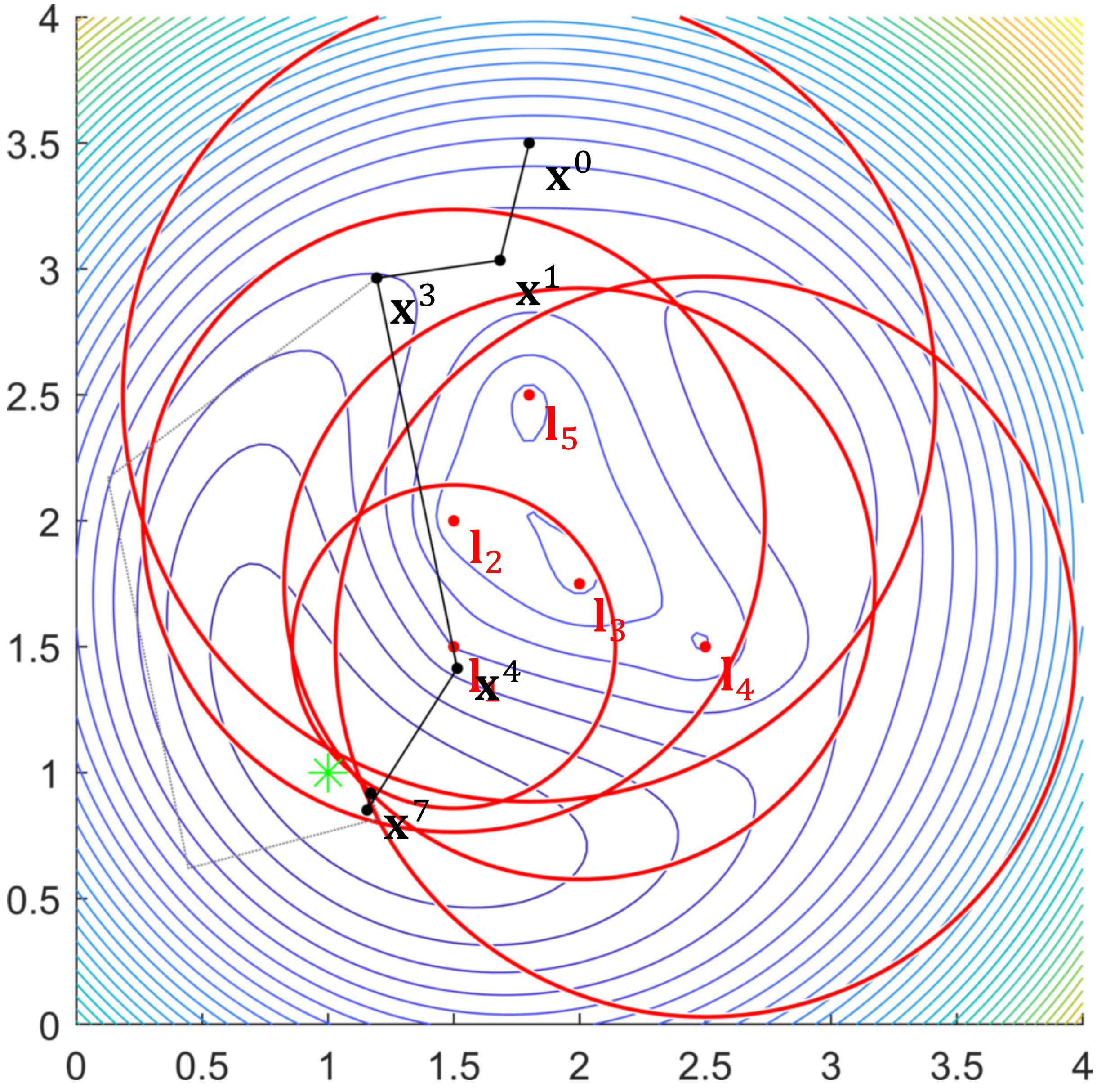
$$X^{t+1} \leftarrow X^{t} + \boldsymbol{\Delta}^{*}$$

- Gauss-Newton
- Levenberg-Marquardt

Example:
Range-based localization

Levenberg–Marquardt optimization





Nonlinear least squares

We can find the MAP estimate
of our unknown states given measurements

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\, p(X \mid Z)$$

by representing it as
a nonlinear least squares problem

$$X^* = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{m} \left\| h_i(X_i) - \mathbf{z}_i \right\|_{\Sigma_i}^2$$

$$\text{Choose a suitable inital estimate } X^0$$

$$\mathbf{A}, \mathbf{b} \leftarrow \text{Linearize at } X^t$$

$$\boldsymbol{\Delta}^* \leftarrow \text{Solve } \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\Delta} - \mathbf{b} \right\|^2$$

$$X^{t+1} \leftarrow X^t + \boldsymbol{\Delta}^*$$

- Uncertainty for MAP estimate by approximating Hessian

Optimizing over poses

- Updates on poses as perturbations in a vector space using **Lie algebra**

$$\mathbf{T} = \exp(\xi^\wedge)\bar{\mathbf{T}}$$

$$\mathfrak{se}(3) = \left\{ \boldsymbol{\Xi} = \boldsymbol{\xi}^\wedge \in \mathbb{R}^{4\times 4} \;\middle|\; \boldsymbol{\xi} \in \mathbb{R}^6 \right\}$$

- Jacobians for these perturbations

$$\left.\frac{\partial\left(\exp(\xi^\wedge)\mathbf{T}\right)\oplus\mathbf{x}}{\partial\xi}\right|_{\xi=\mathbf{0}} = \begin{bmatrix}\mathbf{I}_{3\times 3} & -[\mathbf{T}\oplus\mathbf{x}]^\wedge\end{bmatrix}$$

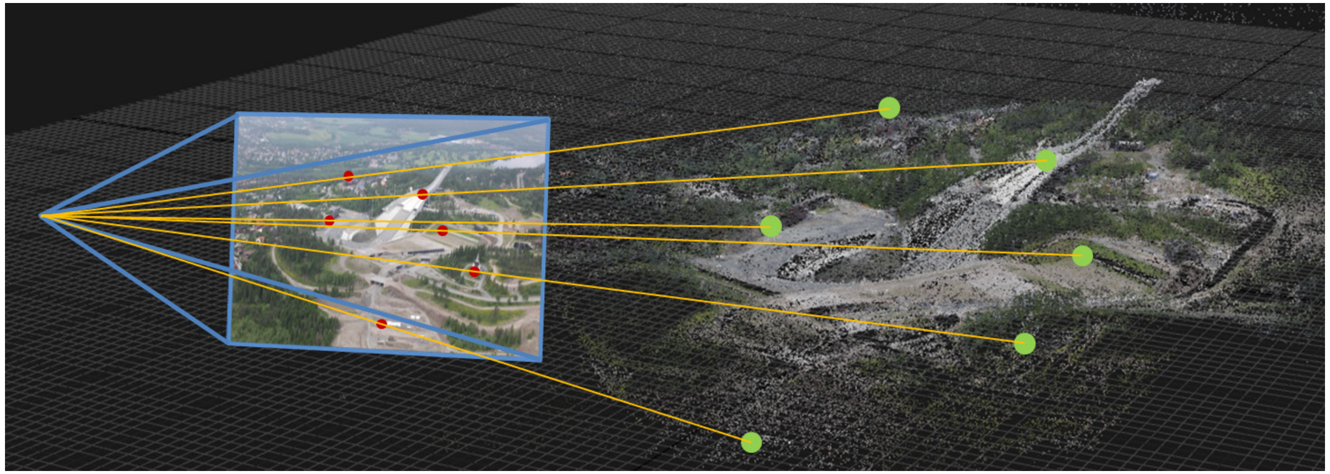
$$\frac{\partial\left(\exp(\xi^\wedge)\mathbf{T}\right)\oplus\mathbf{x}}{\partial\mathbf{x}} = \frac{\partial\mathbf{T}\oplus\mathbf{x}}{\partial\mathbf{x}} = \mathbf{R}$$

The indirect tracking method

Minimize **geometric error** over the **camera pose**

This is also sometimes called **Motion-Only Bundle Adjustment**

$$\mathbf{T}_{cw}^{*} = \underset{\mathbf{T}_{cw}}{\operatorname{argmin}} \sum_{i} \left\| \pi\left(\mathbf{T}_{cw} \tilde{\mathbf{x}}_{i}^{w}\right) - \mathbf{u}_{i} \right\|^{2}$$



Gauss-Newton optimization

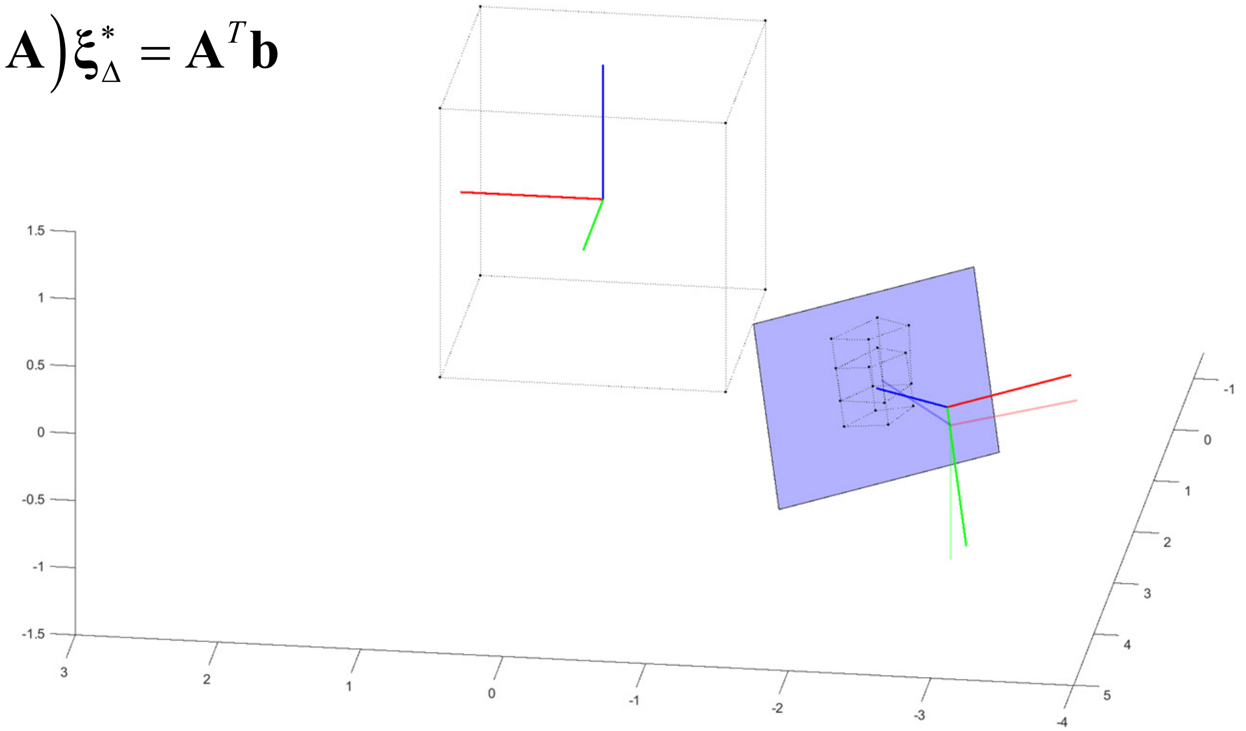
Given a good initial estimate $\mathbf{T}_{wc}^{0}$.

For $t = 0, 1, \dots, t^{max}$

$\mathbf{A}, \mathbf{b} \leftarrow$ Linearize at $\mathbf{T}_{wc}^{t}$

$\boldsymbol{\xi}_{\Delta}^{*} \leftarrow$ Solve the linearized problem with $\left(\mathbf{A}^{T}\mathbf{A}\right)\boldsymbol{\xi}_{\Delta}^{*} = \mathbf{A}^{T}\mathbf{b}$

$\mathbf{T}_{wc}^{t+1} \leftarrow \mathbf{T}_{wc}^{t} \exp(\boldsymbol{\xi}_{\Delta}^{*\wedge})$



Gauss-Newton optimization

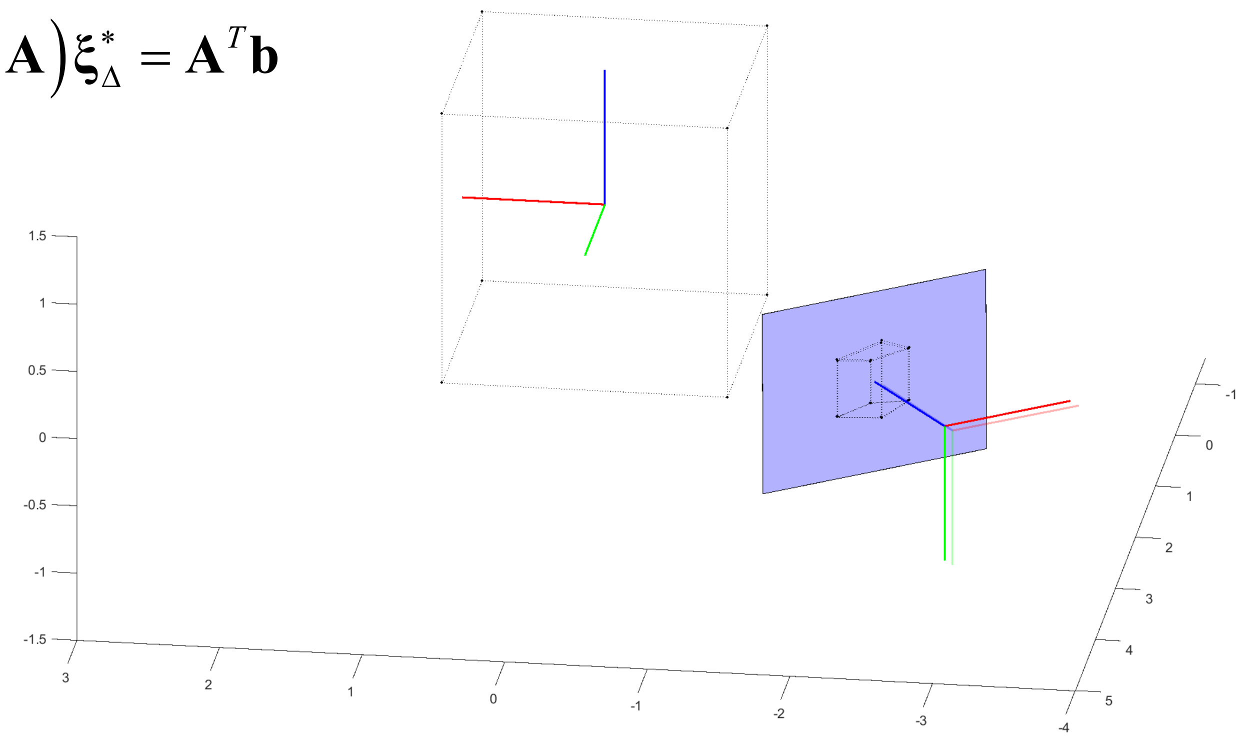
Given a good initial estimate $\mathbf{T}_{wc}^{0}$.

For $t = 0, 1, \dots, t^{max}$

$\mathbf{A}, \mathbf{b} \leftarrow$ Linearize at $\mathbf{T}_{wc}^{t}$

$\boldsymbol{\xi}_{\Delta}^{*} \leftarrow$ Solve the linearized problem with $\left(\mathbf{A}^{T}\mathbf{A}\right)\boldsymbol{\xi}_{\Delta}^{*} = \mathbf{A}^{T}\mathbf{b}$

$\mathbf{T}_{wc}^{t+1} \leftarrow \mathbf{T}_{wc}^{t} \exp(\boldsymbol{\xi}_{\Delta}^{*\wedge})$



Gauss-Newton optimization

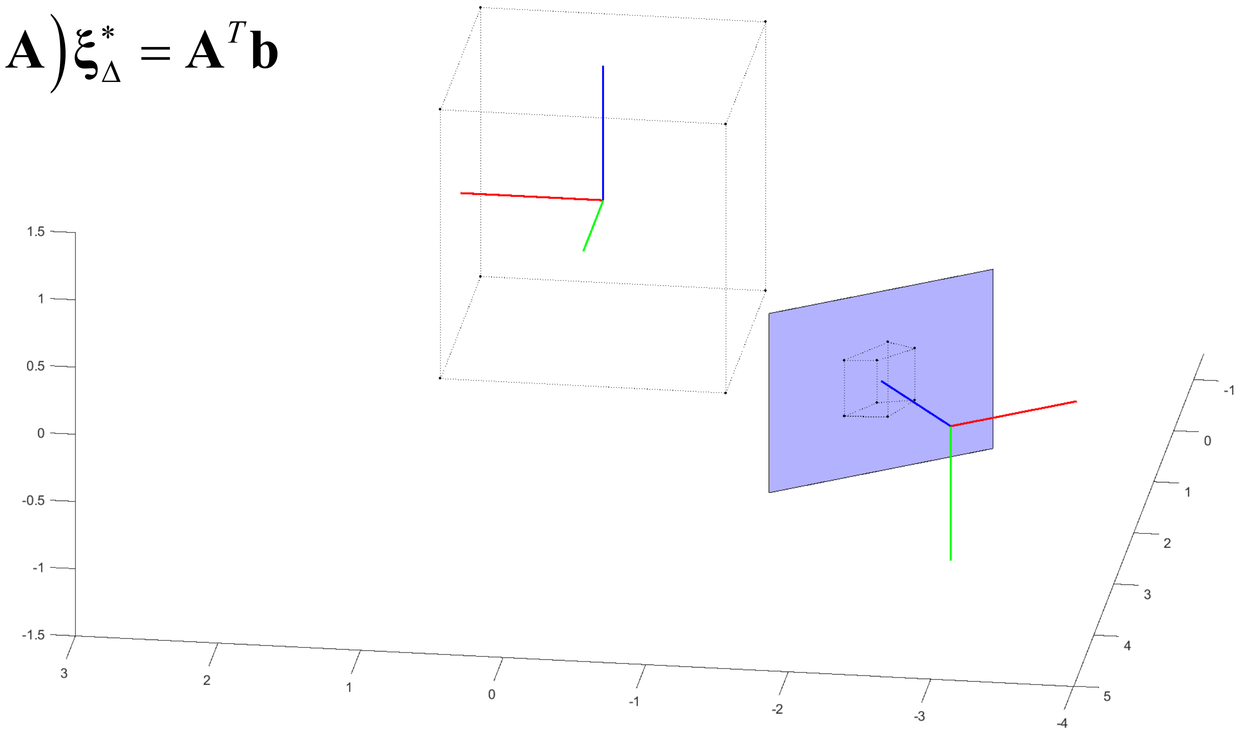
Given a good initial estimate $\mathbf{T}_{wc}^{0}$.

For $t = 0, 1, \dots, t^{max}$

$\mathbf{A}, \mathbf{b} \leftarrow$ Linearize at $\mathbf{T}_{wc}^{t}$

$\boldsymbol{\xi}_{\Delta}^{*} \leftarrow$ Solve the linearized problem with $\left(\mathbf{A}^{T}\mathbf{A}\right)\boldsymbol{\xi}_{\Delta}^{*} = \mathbf{A}^{T}\mathbf{b}$

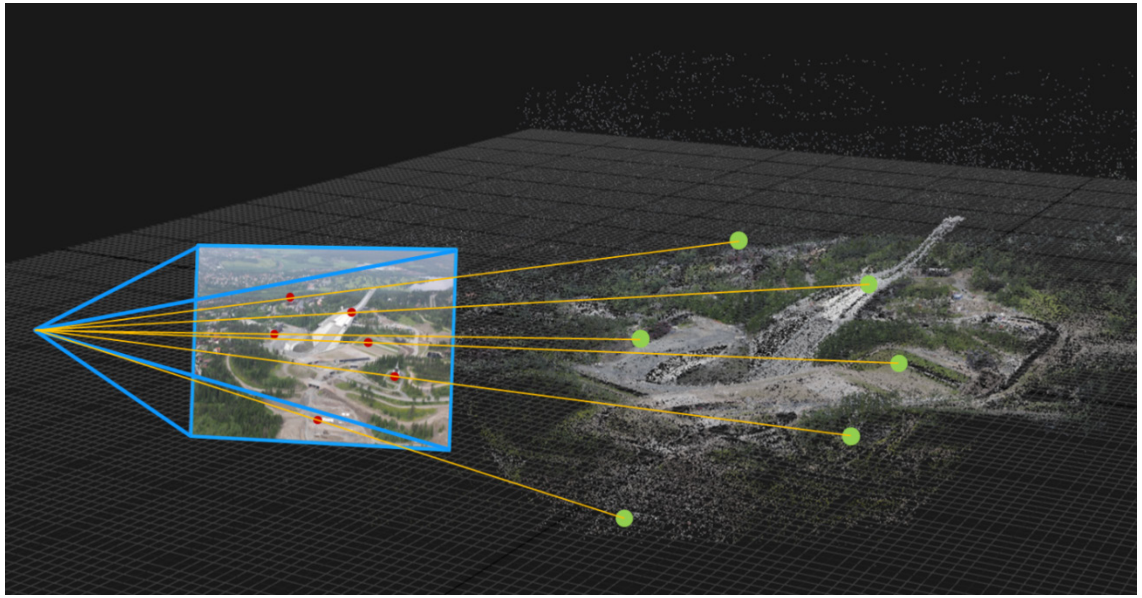
$\mathbf{T}_{wc}^{t+1} \leftarrow \mathbf{T}_{wc}^{t} \exp(\boldsymbol{\xi}_{\Delta}^{*\wedge})$



n-Point Pose Problem (P*n*P)

- Typically fast non-iterative methods
- Minimal in number of points
- Accuracy comparable to iterative methods
- Good for initial estimates

- Examples:
  - P3P, EPnP
  - P4Pf
    - Estimate pose and focal length
  - P6P
    - Estimates **P** with DLT
  - R6P
    - Estimate pose with rolling shutter



Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

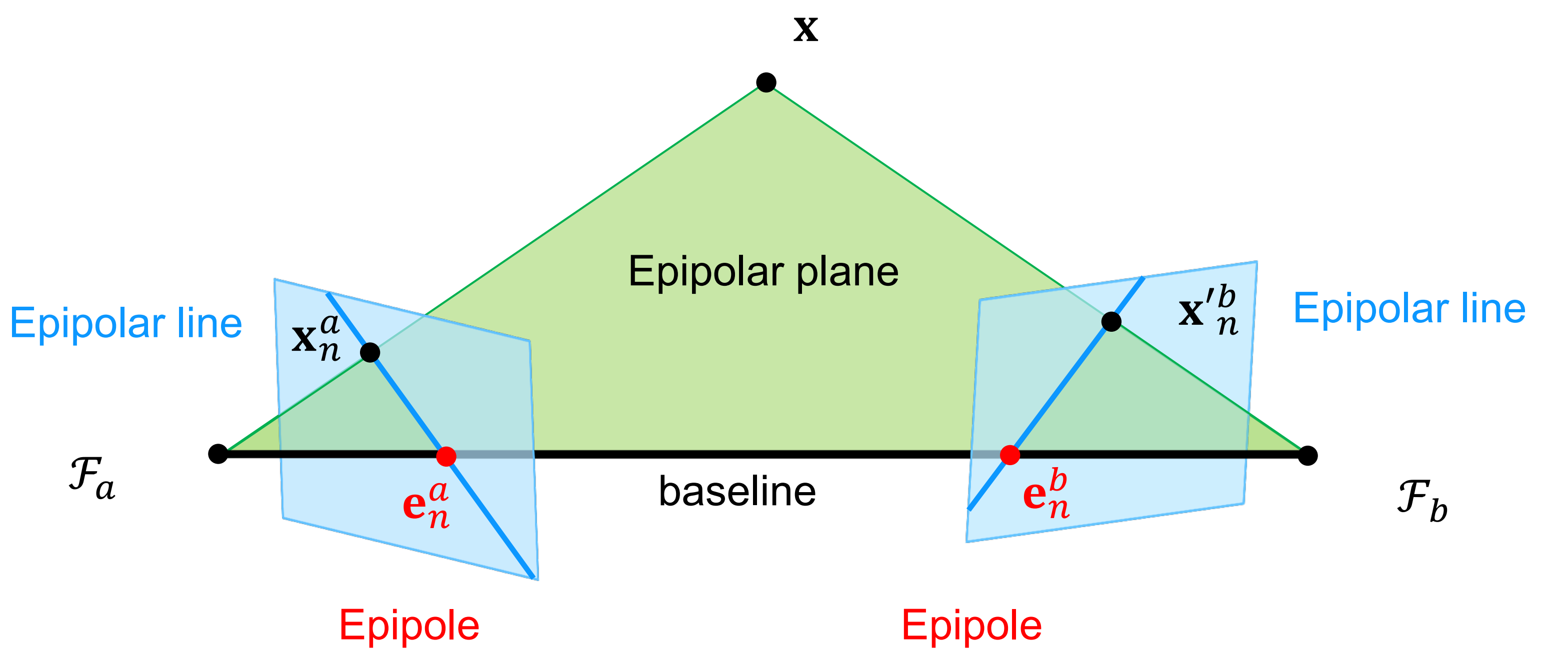
WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Basic epipolar geometry

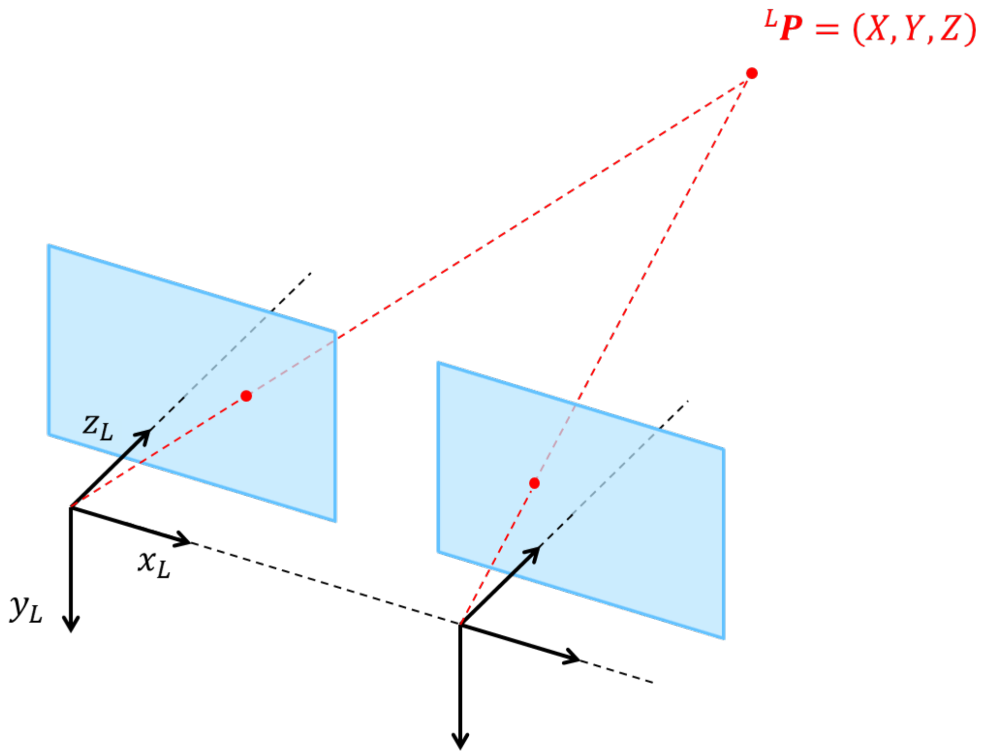


- The **epipolar plane** is the plane containing $\mathbf{x}$ and the two camera centers of $\mathcal{F}_a$ and $\mathcal{F}_b$
- The **baseline** is the line joining $\mathcal{F}_a$ and $\mathcal{F}_b$
- The **epipolar lines** are where the epipolar plane intersect the image planes
- The **epipoles** are where the baseline intersects the two image planes
- Epipoles and epipolar lines can be represented in the normalized image plane as well as in the image

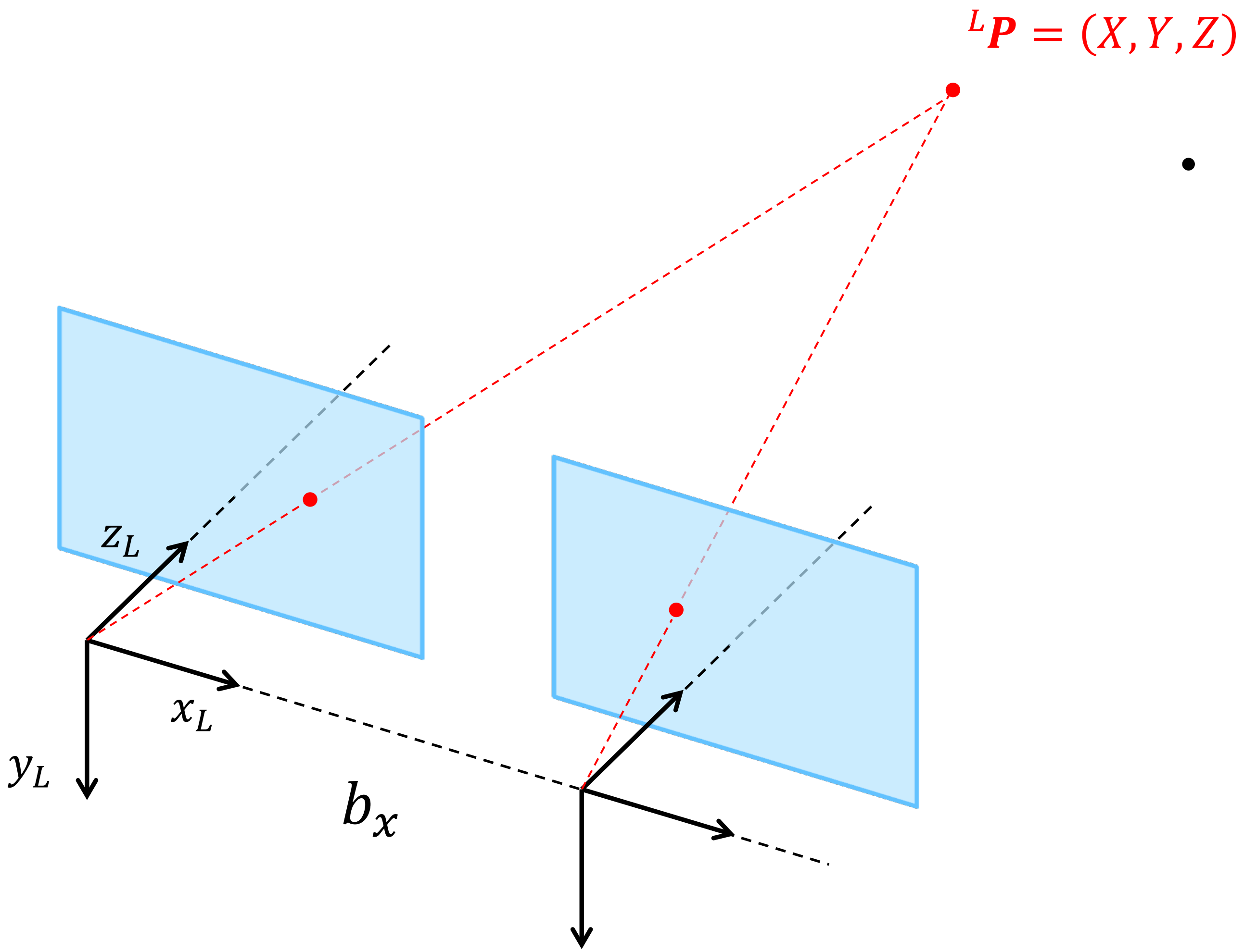
Stereo imaging

Stereo imaging

- Stereo imaging
  - Horizontal epipolar lines
  - Disparity
  - 3D from disparity
  - Stereo rectification

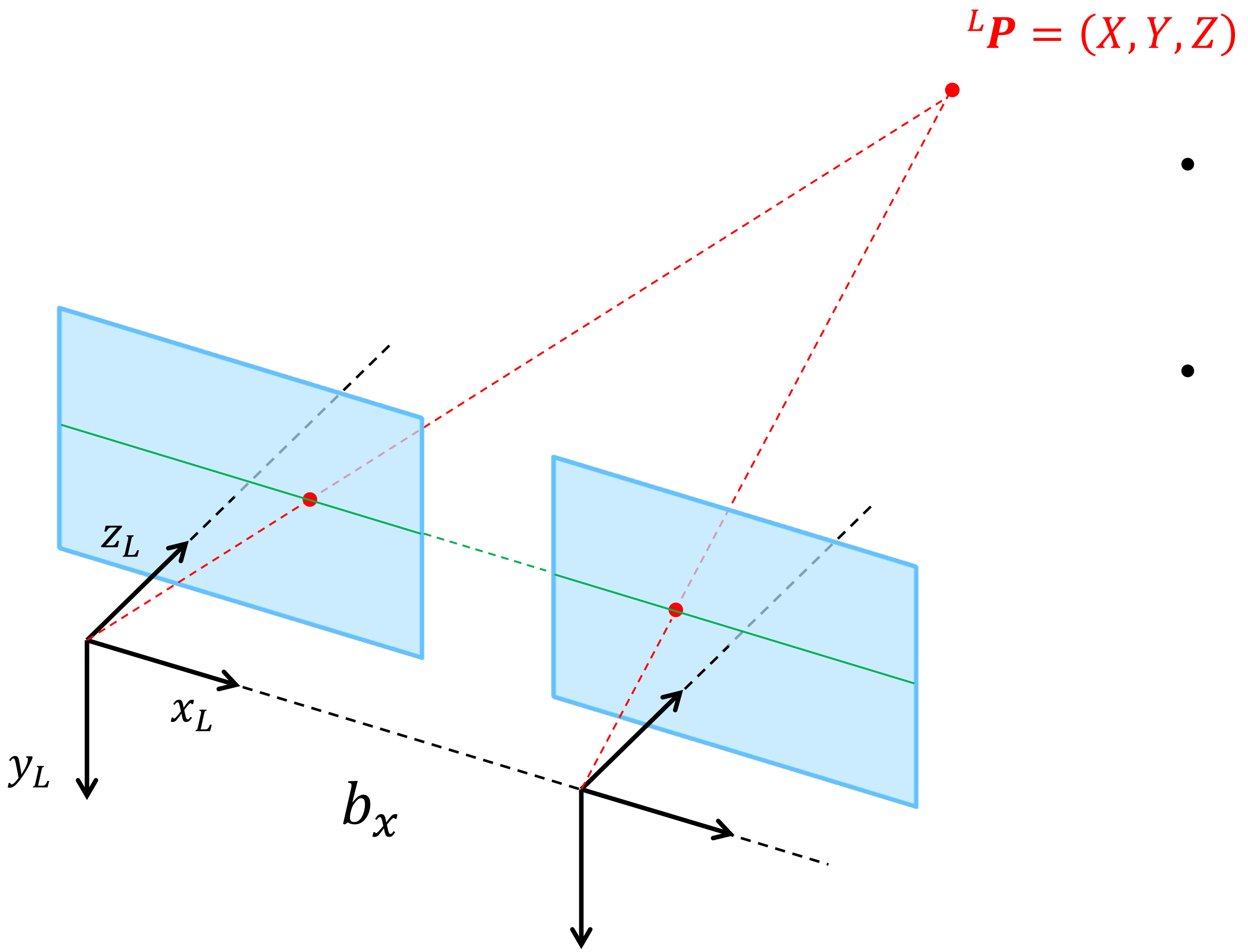


Stereo geometry



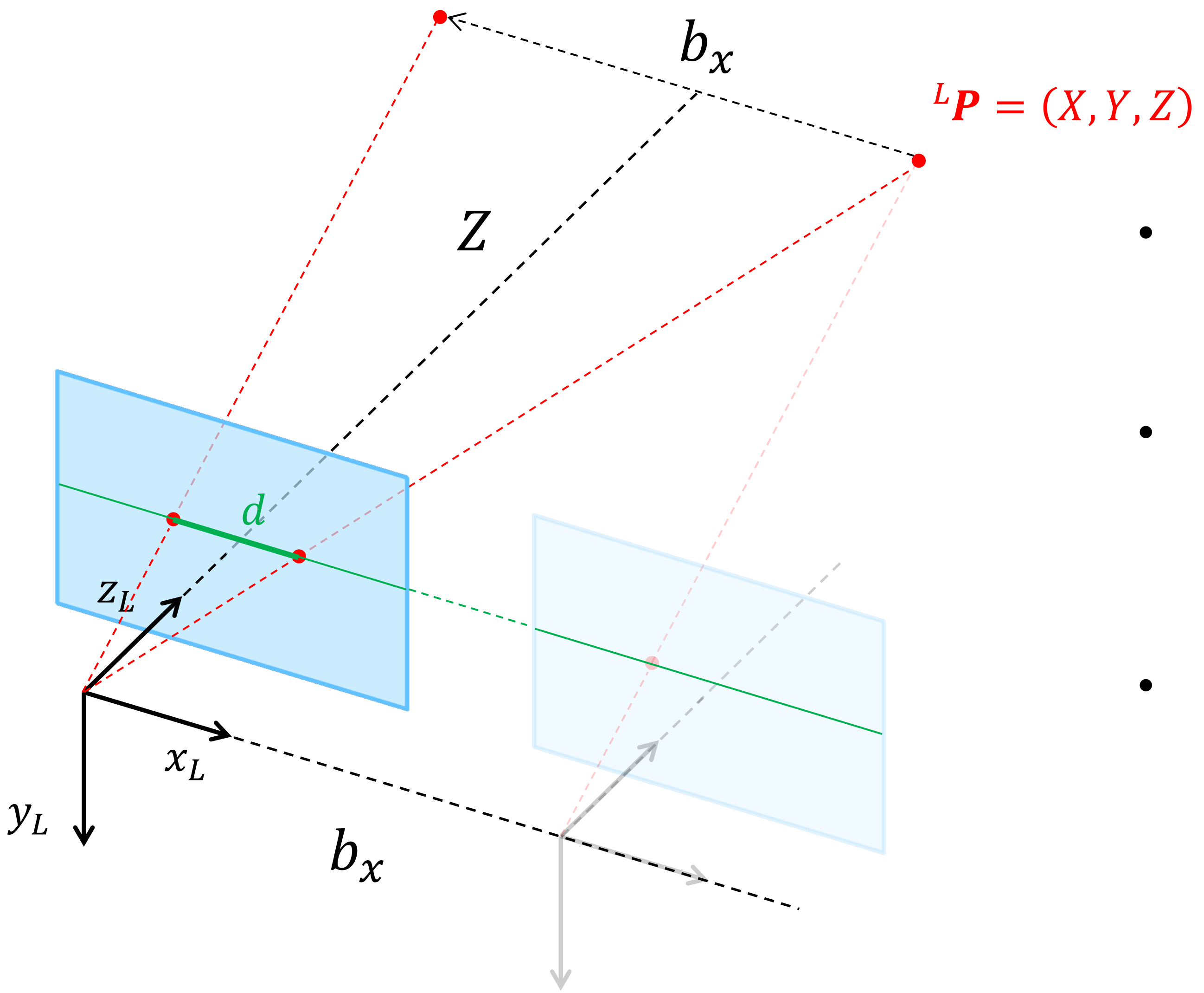
- Parallel identical cameras
  - Translated along *x*-axis

Stereo geometry



- Parallel identical cameras
  - Translated along *x*-axis
- Horizontal epipolar lines
  - Corresponding points lie along the same row in the two images

Stereo geometry

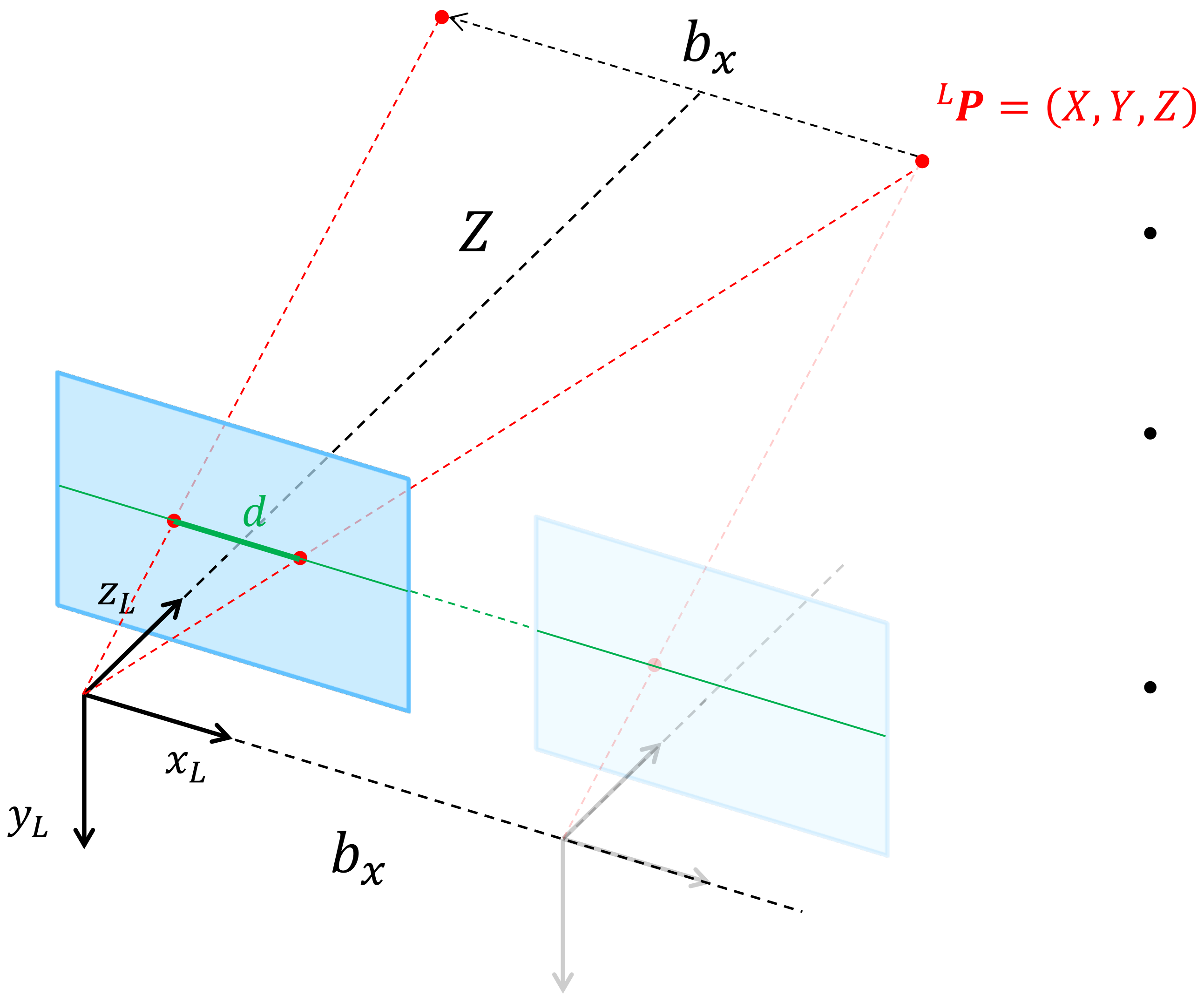


- Parallel identical cameras
  - Translated along *x*-axis
- Horizontal epipolar lines
  - Corresponding points lie along the same row in the two images
- Depth from disparity

$$Z = f\frac{b_x}{d}$$

Depth: Z; Baseline: b_x; Disparity: d

Stereo geometry

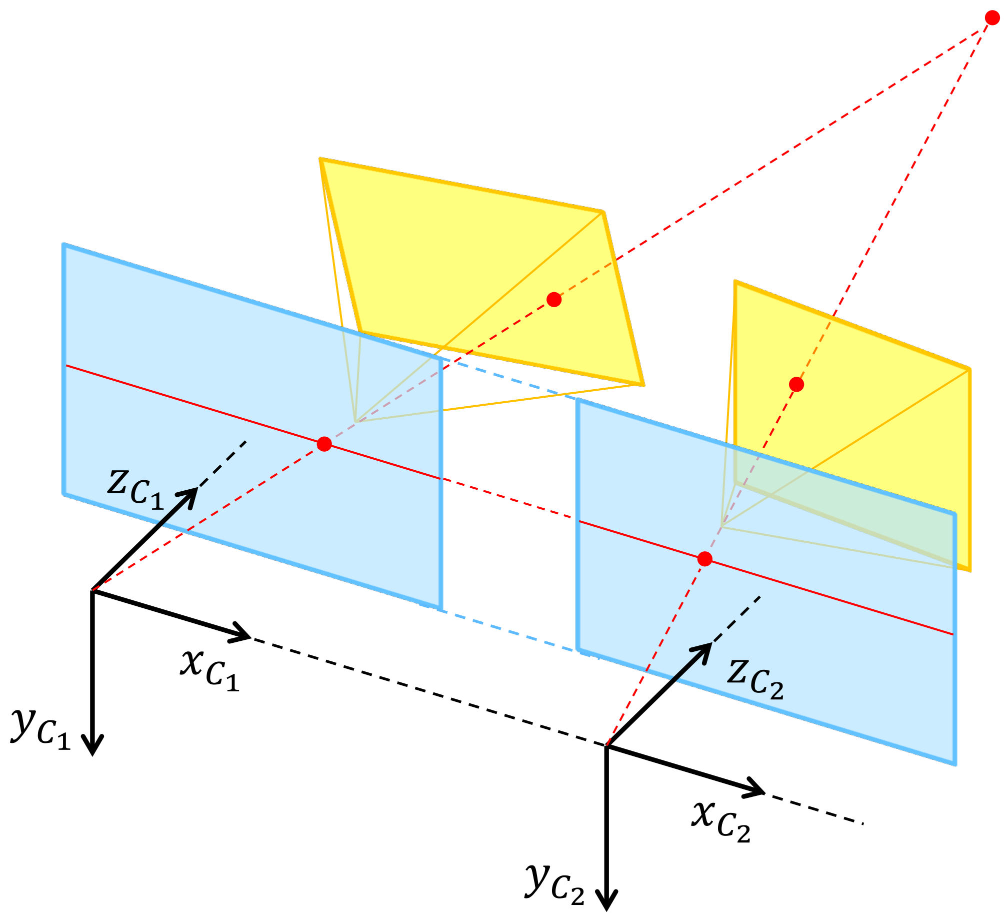


- Parallel identical cameras
  - Translated along *x*-axis
- Horizontal epipolar lines
  - Corresponding points lie along the same row in the two images
- 3D from disparity

Depth, Baseline, Disparity

$$Z = f\frac{b_x}{d} \qquad X = x_L\frac{b_x}{d} \qquad Y = y_L\frac{b_x}{d}$$

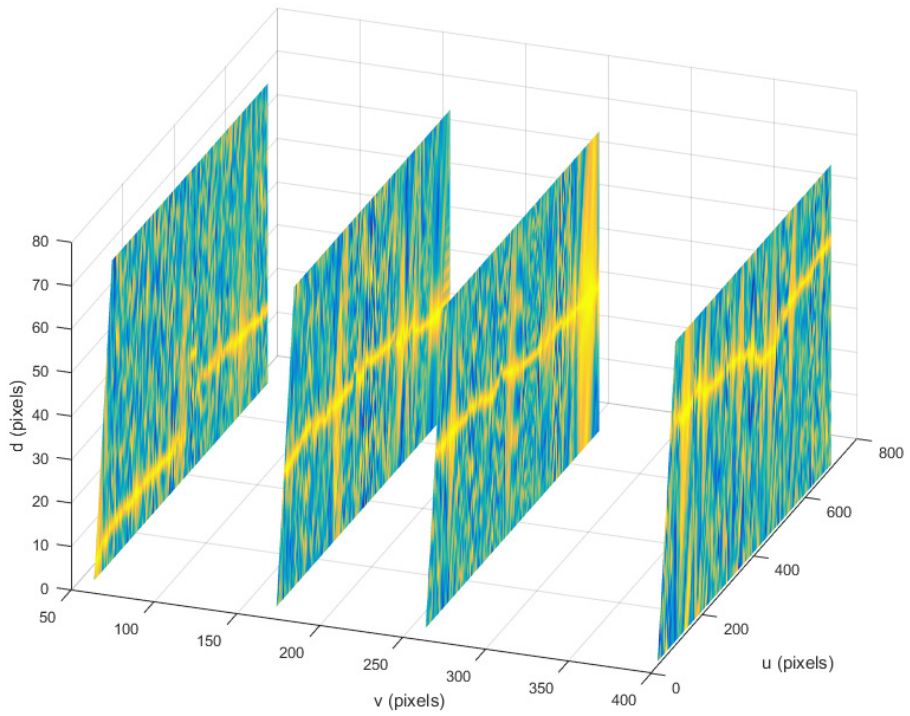
Stereo rectification



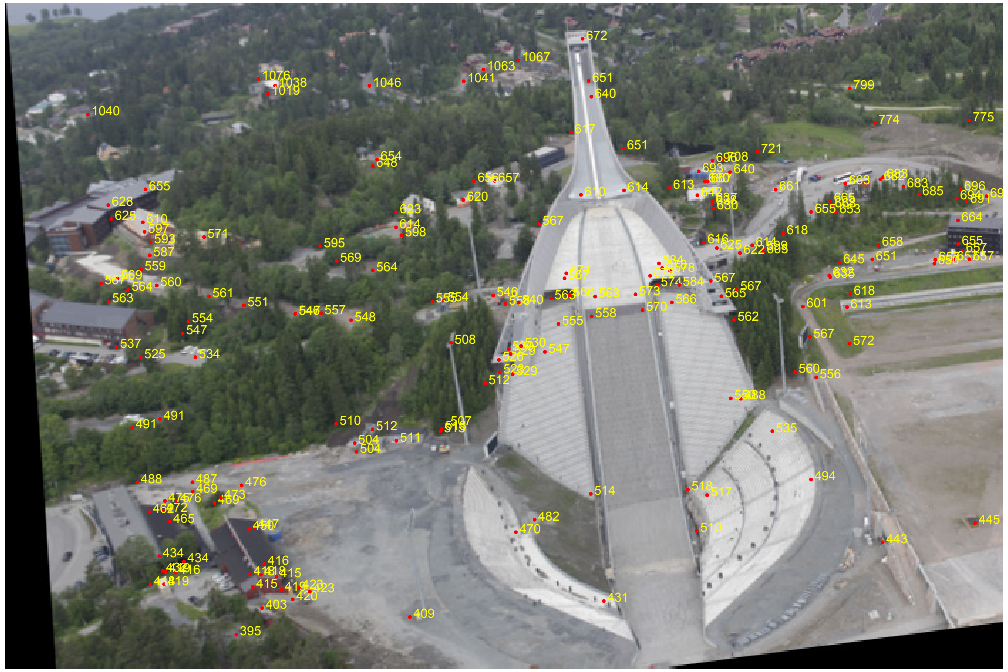
- Reproject image planes onto a common plane parallel to the line between the camera centers
- The epipolar lines are horizontal after this transformation
- Two homographies
- C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Stereo imaging
Stereo processing

- Stereo processing
  - Sparse vs dense matching
  - DSI
  - Typical failures
  - Removing failures vs smoothness



Stereo processing



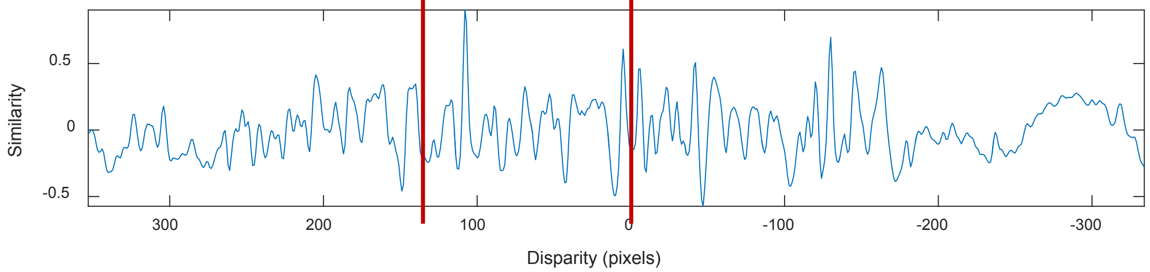
- Sparse stereo
  - Extract keypoints
  - Match keypoints along the same row
  - Compute 3D from disparity

- Dense stereo
  - Try to match all pixels along rows
  - Compute disparity image by finding the best disparity for each pixel
  - Refine and clean disparity image
  - Compute dense 3D point cloud or surface from disparity

Dense stereo matching

- For a patch in the left image
  - Compare with patches along the same row in the right image
  - Select patch with highest score

- Repeat for all pixels in the left image



Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Representing the epipolar geometry

- The essential matrix $\mathbf{E}$ and the fundamental matrix $\mathbf{F}$ represent the epipolar geometry

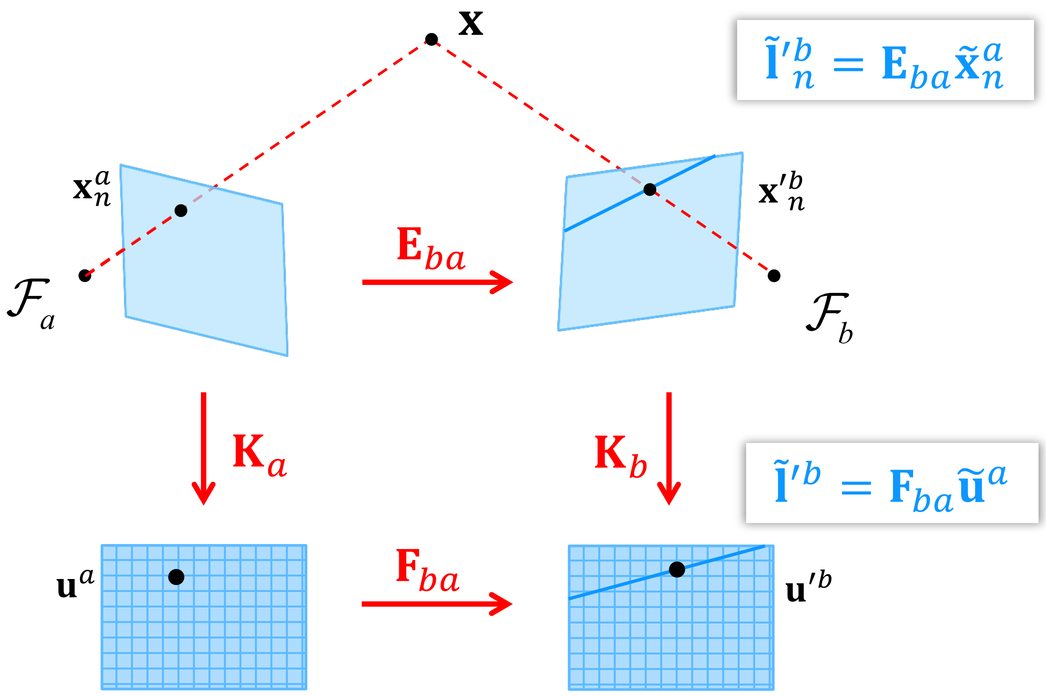
$$\left(\tilde{\mathbf{x}}_n^{\prime b}\right)^T \mathbf{E}_{ba} \tilde{\mathbf{x}}_n^a = 0 \qquad \left(\tilde{\mathbf{u}}^{\prime b}\right)^T \mathbf{F}_{ba} \tilde{\mathbf{u}}^a = 0$$

- $\mathbf{E}$ and $\mathbf{F}$ can be estimated from point correspondences
  - $\mathbf{F} \leftarrow$ RANSAC, 7-pt or 8-pt
  - $\mathbf{E} \leftarrow$ RANSAC, 5-pt

- $\mathbf{E}$ and $\mathbf{F}$ maps points to epipolar lines

- The essential matrix is related directly to the relative pose between the two cameras

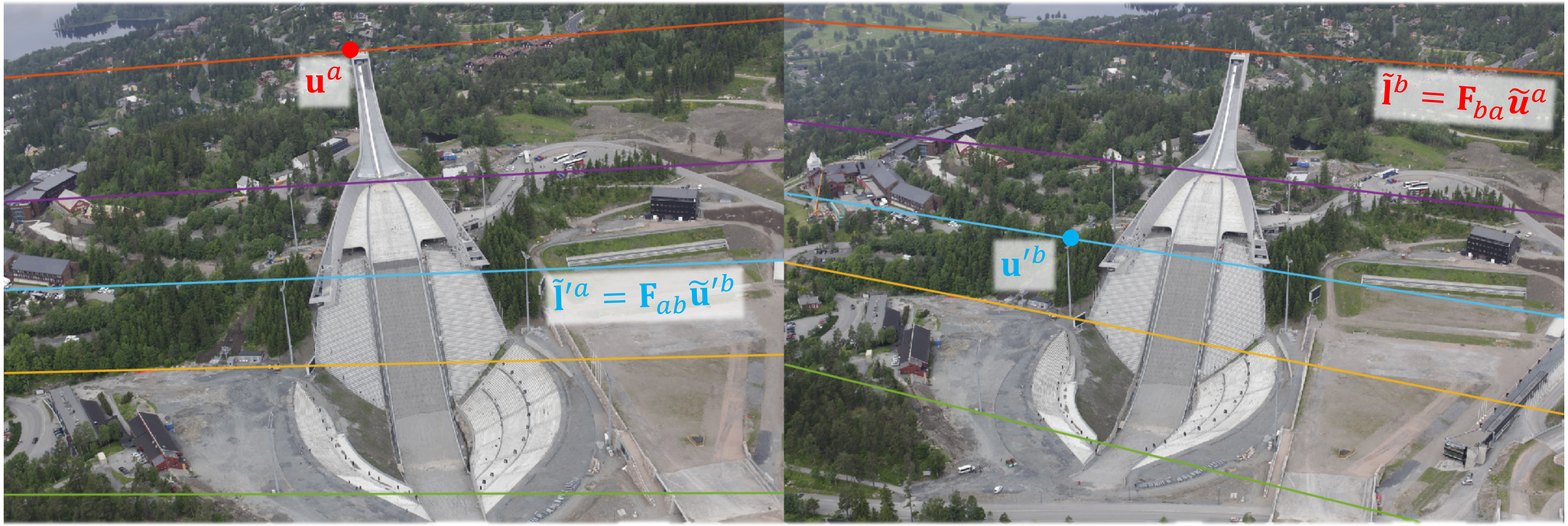
$$\mathbf{E}_{ba} = \left(\mathbf{t}_{ba}^b\right)^\wedge \mathbf{R}_{ba}$$



$$\mathbf{F}_{ba} = \mathbf{K}_b^{-T} \mathbf{E}_{ba} \mathbf{K}_a^{-1}$$

$$\mathbf{F}_{ab} = \mathbf{K}_a^{-T} \mathbf{E}_{ab} \mathbf{K}_b^{-1}$$

Example



img_a img_b

Linear triangulation by minimizing the algebraic error

Assume that we know the camera projection matrices $\mathbf{P}_a$, $\mathbf{P}_b$ and a 2D correspondence $\mathbf{u}^a \leftrightarrow \mathbf{u}'^b$ for a 3D point $\mathbf{x}$

$$\tilde{\mathbf{u}}^a = \mathbf{P}_a \tilde{\mathbf{x}}$$

$$\begin{bmatrix} u^a \\ v^a \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_a^{1T} \\ \mathbf{p}_a^{2T} \\ \mathbf{p}_a^{3T} \end{bmatrix} \tilde{\mathbf{x}}$$

$$\Downarrow$$

$$\begin{bmatrix} v^a \mathbf{p}_a^{3T} - \mathbf{p}_a^{2T} \\ \mathbf{p}_a^{1T} - u^a \mathbf{p}_a^{3T} \end{bmatrix} \tilde{\mathbf{x}} = \mathbf{0}$$

$$\tilde{\mathbf{u}}'^b = \mathbf{P}_b \tilde{\mathbf{x}}$$

$$\begin{bmatrix} u'^b \\ v'^b \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_b^{1T} \\ \mathbf{p}_b^{2T} \\ \mathbf{p}_b^{3T} \end{bmatrix} \tilde{\mathbf{x}}$$

$$\Downarrow$$

$$\begin{bmatrix} v'^b \mathbf{p}_b^{3T} - \mathbf{p}_b^{2T} \\ \mathbf{p}_b^{1T} - u'^b \mathbf{p}_b^{3T} \end{bmatrix} \tilde{\mathbf{x}} = \mathbf{0}$$

Each perspective camera model gives rise to two equations on the three entries of $\mathbf{x}$

Combining these equations gives us an overdetermined homogenous system of linear equations that we can solve with SVD to find the 3D point $\mathbf{x}$ that minimize the **algebraic error**

$$\varepsilon = \left\| \mathbf{A}\tilde{\mathbf{x}} \right\|$$

in a linear least squares sense

$$\begin{bmatrix} v^a \mathbf{p}_a^{3T} - \mathbf{p}_a^{2T} \\ \mathbf{p}_a^{1T} - u^a \mathbf{p}_a^{3T} \\ v'^b \mathbf{p}_b^{3T} - \mathbf{p}_b^{2T} \\ \mathbf{p}_b^{1T} - u'^b \mathbf{p}_b^{3T} \end{bmatrix} \tilde{\mathbf{x}} = \mathbf{0}$$

$$\mathbf{A}\tilde{\mathbf{x}} = \mathbf{0}$$

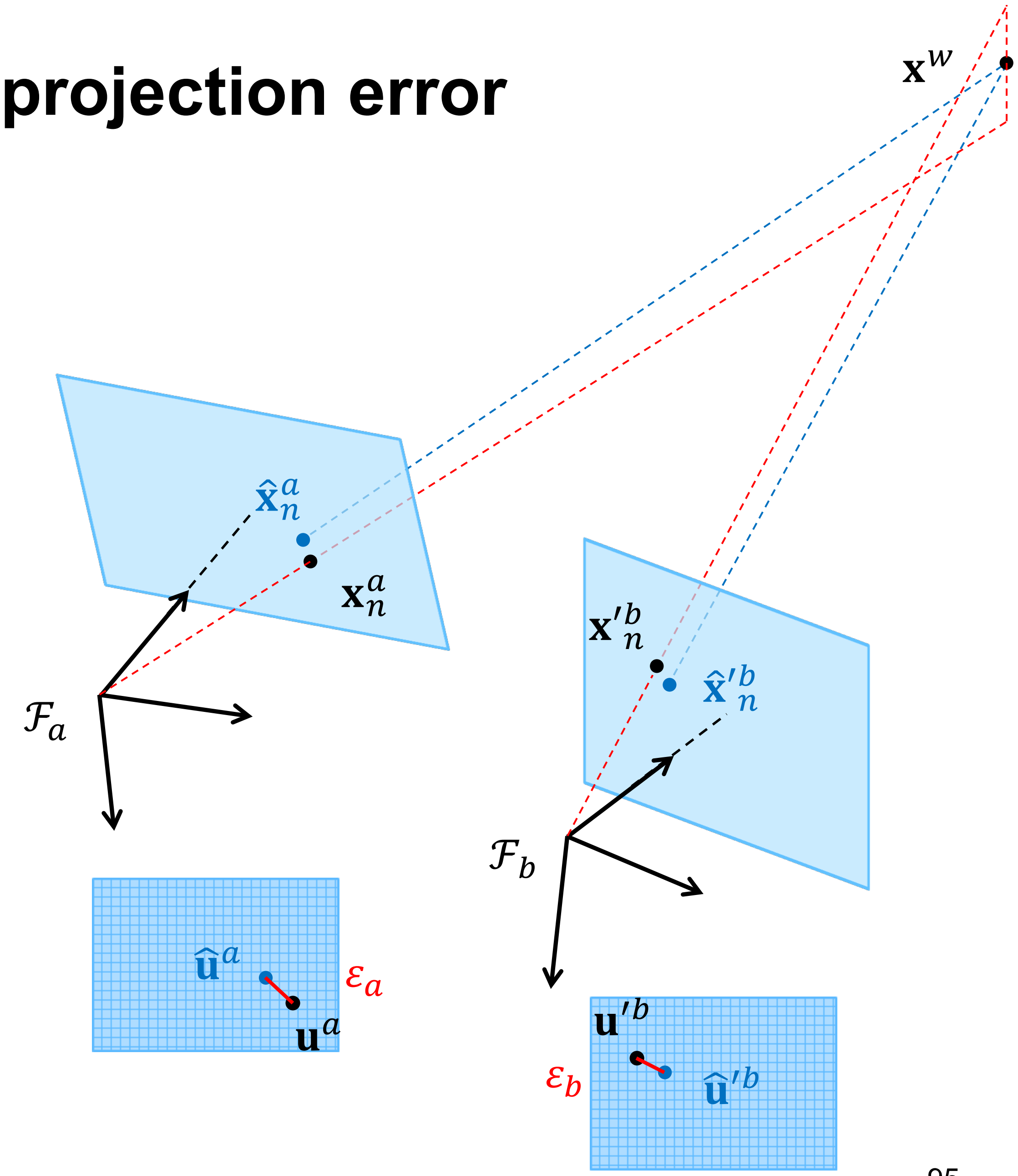
Triangulation by minimizing the reprojection error

If we denote the camera projections by π_a and π_b, then the **reprojection error** ε is given by

$$\varepsilon = \varepsilon_a^{\,2} + \varepsilon_b^{\,2}$$

$$= \left\| \pi_a\left(\mathbf{T}_{aw}\tilde{\mathbf{x}}^w\right) - \mathbf{u}^a \right\|^2 + \left\| \pi_b\left(\mathbf{T}_{bw}\tilde{\mathbf{x}}^w\right) - \mathbf{u}^b \right\|^2$$

Estimating $\tilde{\mathbf{x}}^w$ by minimizing ε is a non-linear optimization problem, which needs an initial estimate

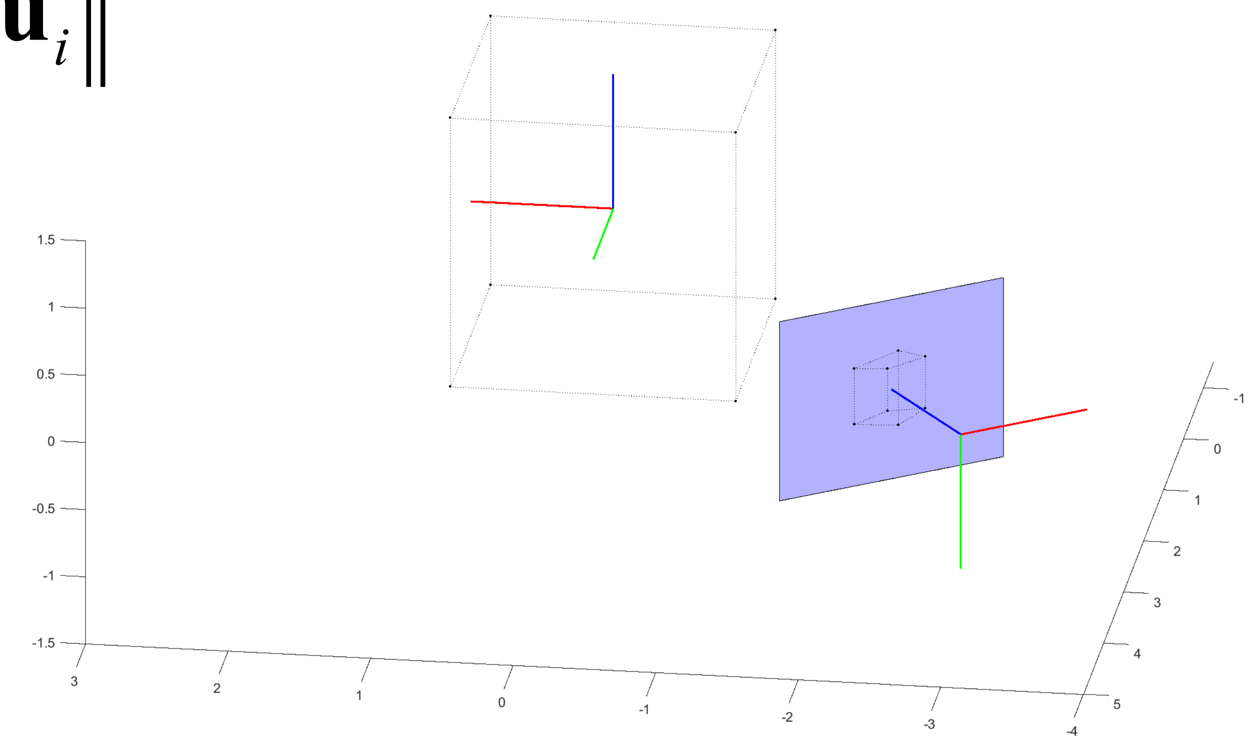


Pose estimation by minimizing reprojection error

Minimize **geometric error** over the **camera pose**

This is also sometimes called **Motion-Only Bundle Adjustment**

$$\mathbf{T}_{cw}^{*} = \underset{\mathbf{T}_{cw}}{\operatorname{argmin}} \sum_{i} \left\| \pi(\mathbf{T}_{cw}\tilde{\mathbf{x}}_{i}^{w}) - \mathbf{u}_{i} \right\|^{2}$$

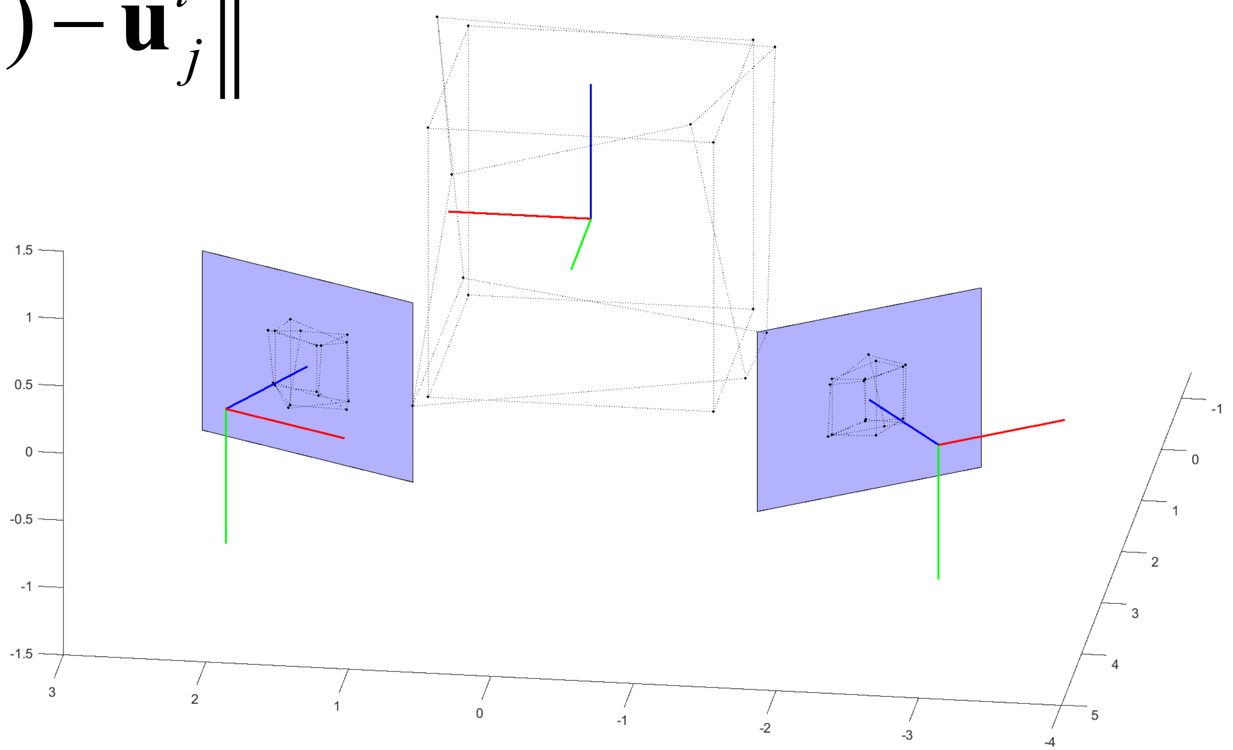


Triangulation by minimizing reprojection error

Minimize **geometric error** over the **world points**

This is also sometimes called **Structure-Only Bundle Adjustment**

$$\mathbf{x}_j^{w*} = \underset{\mathbf{x}_j^{w*}}{\operatorname{argmin}} \sum_i \sum_j \left\| \pi_i(\mathbf{T}_{cw_i} \tilde{\mathbf{x}}_j^w) - \mathbf{u}_j^i \right\|^2$$



Two-view geometry
Pose from epipolar geometry

- Non-planar case
  - Estimate epipolar geometry
  - Estimate relative pose from E

- Planar case
  - Estimate homography
  - Estimate relative pose from H

Pose from epipolar geometry

There are four different poses that satisfy the equation
$\mathbf{E}_{ba} = \left(\mathbf{t}_{ba}^{b}\right)^{\wedge}\mathbf{R}_{ba}$

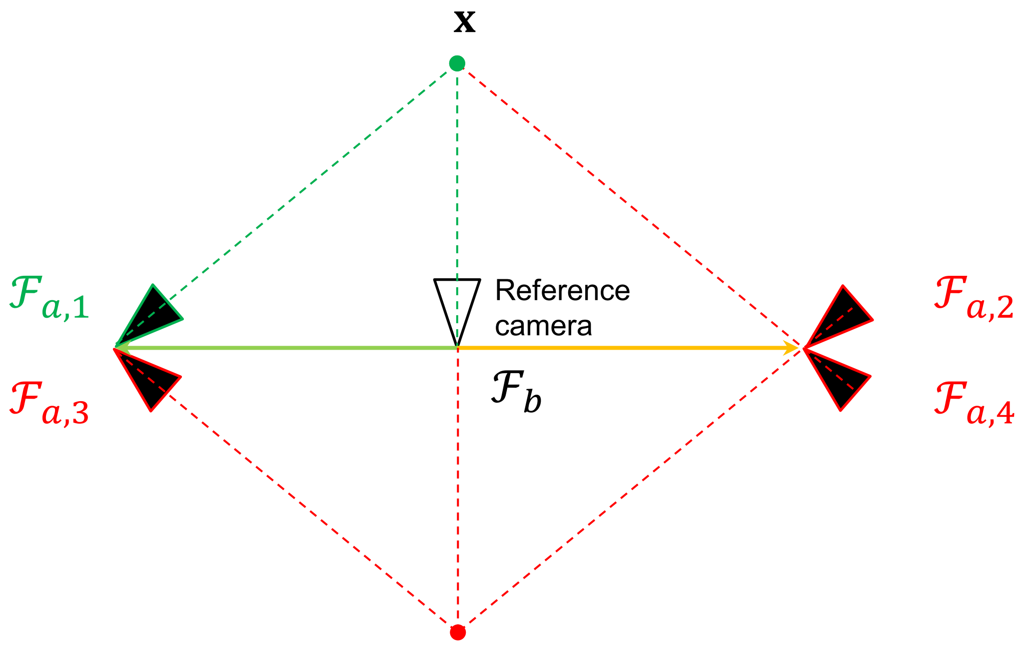
The figure illustrates how this might look like for the case when $\mathbf{T}_{ba,1}$ is the correct pose

$\mathbf{T}_{ba,i}$ is the pose of $\mathcal{F}_{a,i}$ relative to $\mathcal{F}_b$

There is no way of predicting the correct pose out of the four, but in general only one of them corresponds to $\mathbf{x}$ being in front of both cameras

This constraint is known as **the chirality constraint** and it is tested by triangulation of at least one 3D point

$\left\|\mathbf{t}_{ba}^{b}\right\|$ can <u>not</u> be found from $\mathbf{E}_{ba}$ (homogeneous matrix)

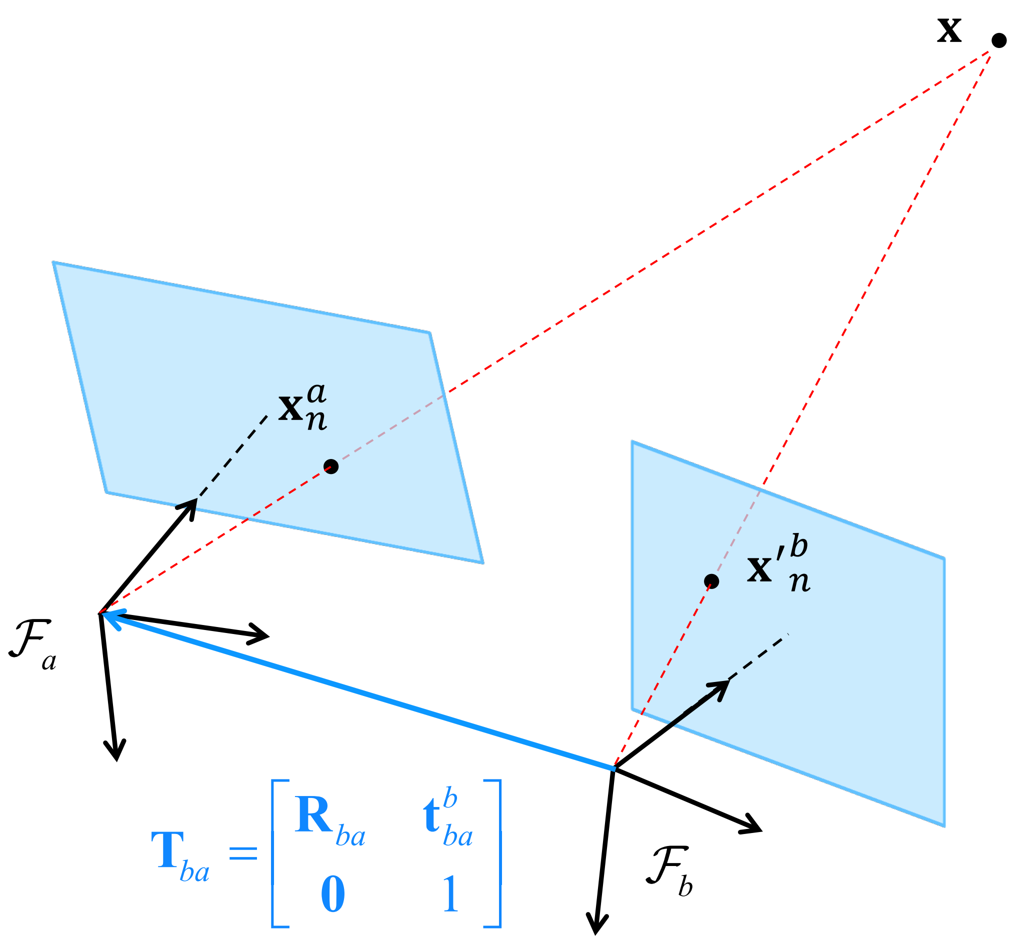


Pose from epipolar geometry

Pose between two calibrated cameras

1. Establish robust correspondences $\mathbf{u}_i^a \leftrightarrow \mathbf{u}'^b_i$ between images
2. Determine coorspondences $\mathbf{x}_{n,i}^a \leftrightarrow \mathbf{x}'^b_{n,i}$ using that $\tilde{\mathbf{x}}_n = \mathbf{K}^{-1}\tilde{\mathbf{u}}$
3. Estimate the essential matrix $\mathbf{E}_{ba}$ from correspondences $\mathbf{x}_{n,i}^a \leftrightarrow \mathbf{x}'^b_{n,i}$
4. Compute poses $\mathbf{T}_{ba,1}, \dots, \mathbf{T}_{ba,4}$ from $\mathbf{E}_{ba}$
5. For each pose, determine at least one 3D point $\mathbf{x}$ by triangulation and select the pose that satisfies the chirality constraint

$\left\|\mathbf{t}_{ba}^b\right\|$ remains unknown!



Planar scene

One can prove that if

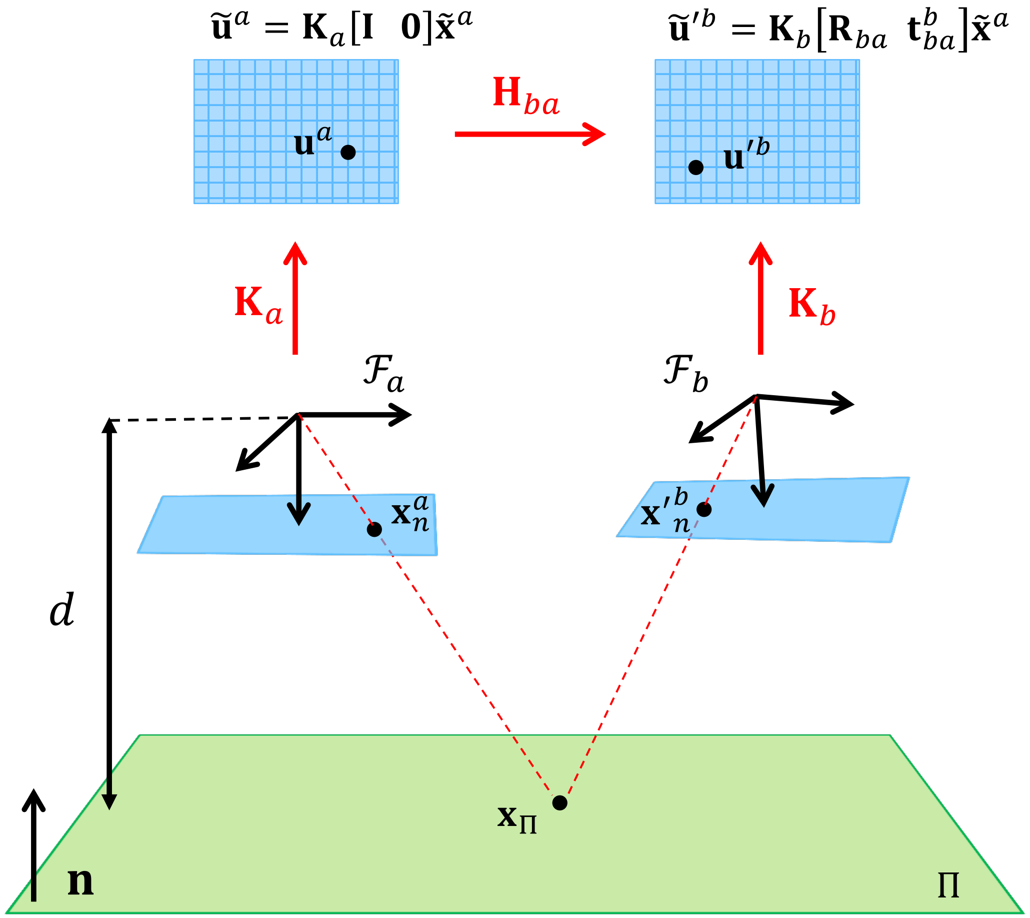
$$\mathbf{T}_{ba} = \begin{bmatrix} \mathbf{R}_{ba} & \mathbf{t}_{ba}^{b} \\ \mathbf{0} & 1 \end{bmatrix}$$

then

$$\mathbf{H}_{ba} = \mathbf{K}_b\left(\mathbf{R}_{ba} - \mathbf{t}_{ba}^{b}(\mathbf{n}^a)^T/d\right)\mathbf{K}_a^{-1}$$

It is possible to estimate $\left(\mathbf{R}_{ba}, \mathbf{n}^a, \frac{1}{d}\mathbf{t}_{ba}^{b}\right)$ from a known homography

– Four solutions



Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

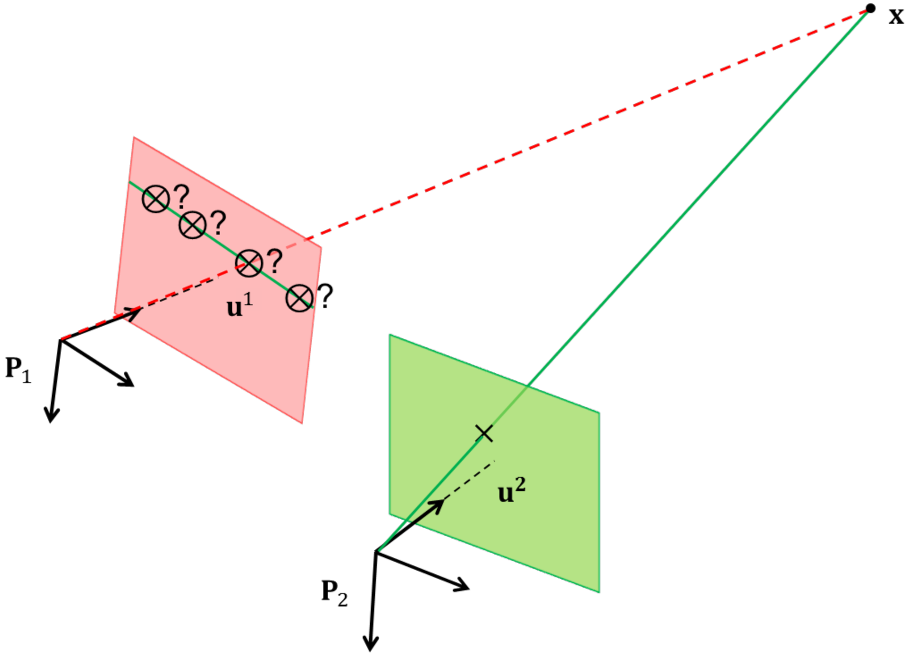
WORLD GEOMETRY AND 3D

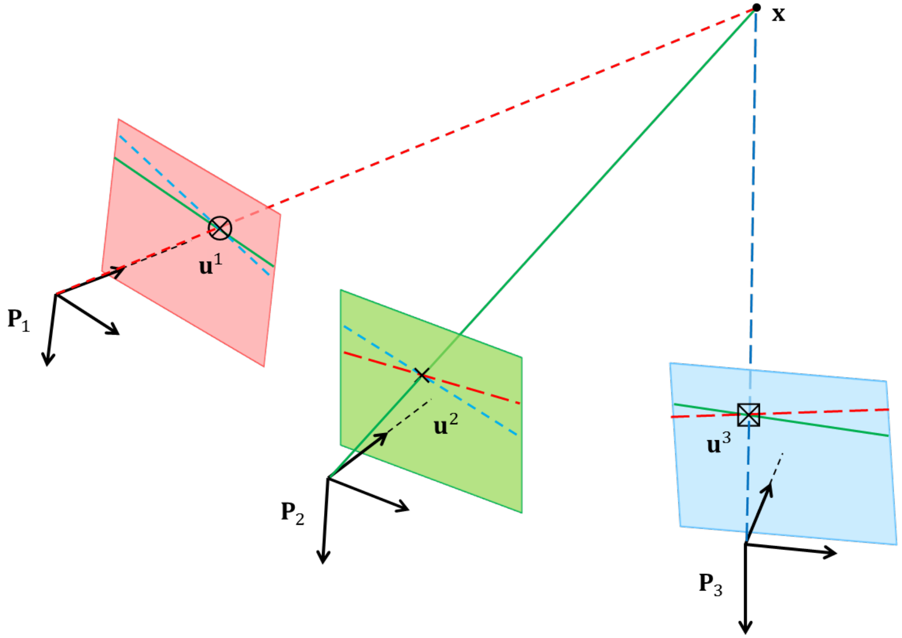
- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

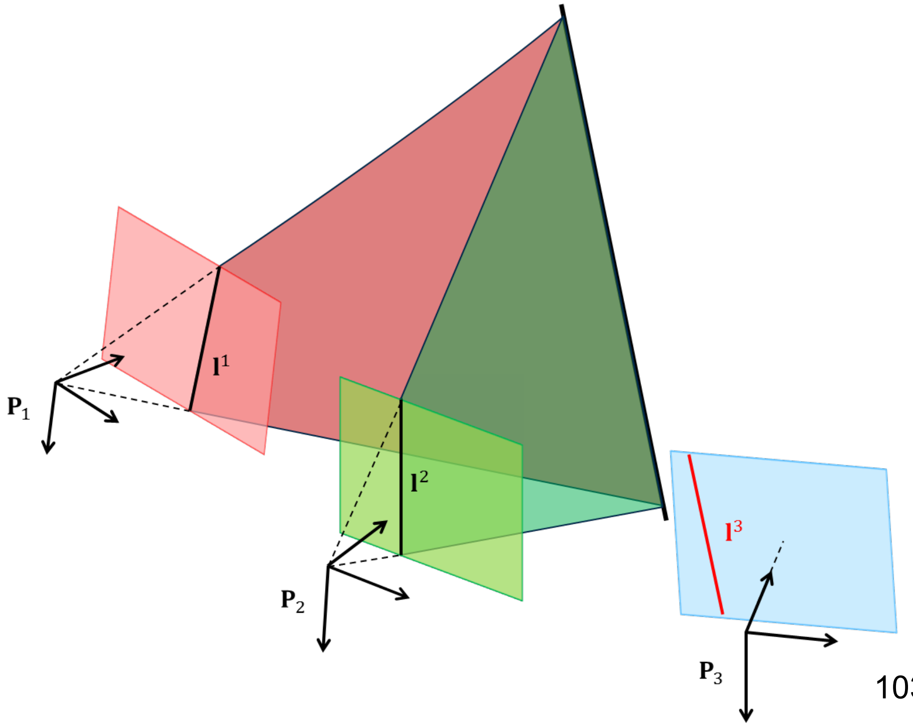
SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Multiple-view geometry







- Multiple-view geometry
- Correspondences
  - Two-view vs Three-view
  - Fundamental matrix vs Trifocal tensor

Multiple-view geometry

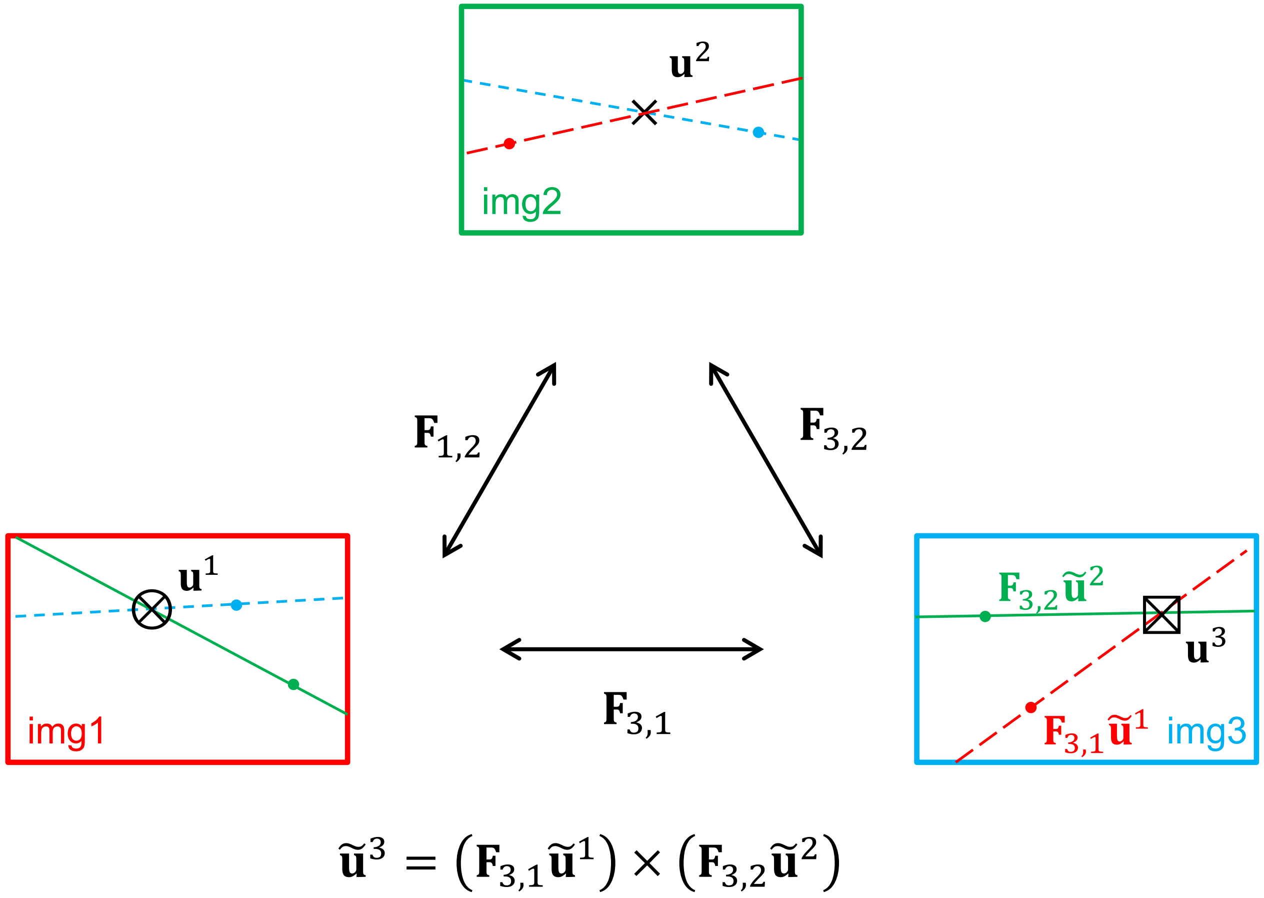
Three views

- Given three overlapping images, we can establish (or evaluate) point correspondences using the pairwise epipolar constraints

$$\tilde{\mathbf{u}}^3 = \left(\mathbf{F}_{3,1}\tilde{\mathbf{u}}^1\right) \times \left(\mathbf{F}_{3,2}\tilde{\mathbf{u}}^2\right)$$

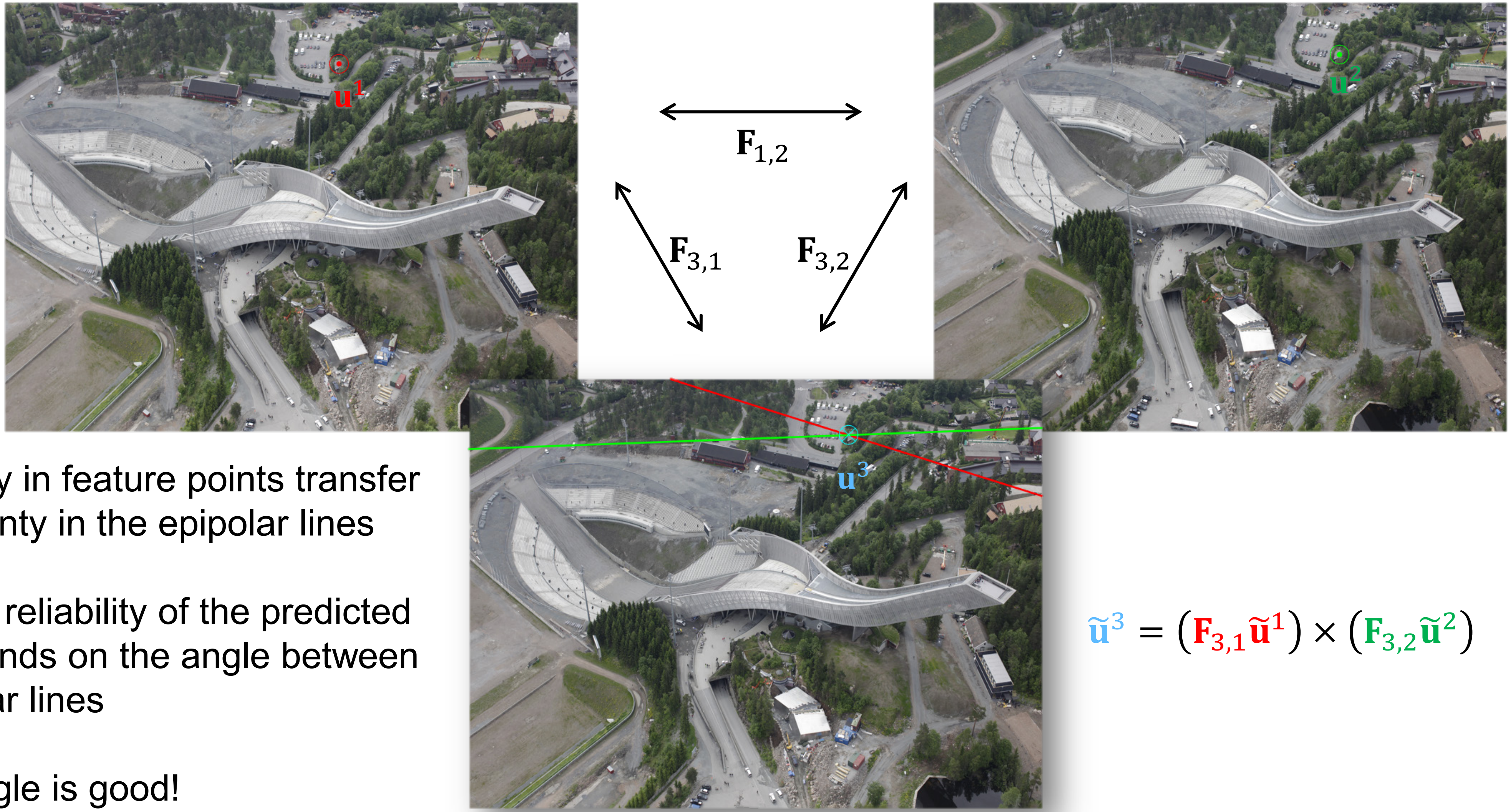
- However, this fails for points in the plane defined by the three camera centers – the trifocal plane – since the epipolar lines then will coincide

- The trifocal tensor allows point transfer also for points in the trifocal plane



Example

Point transfer based on epipolar constraints



Uncertainty in feature points transfer to uncertainty in the epipolar lines

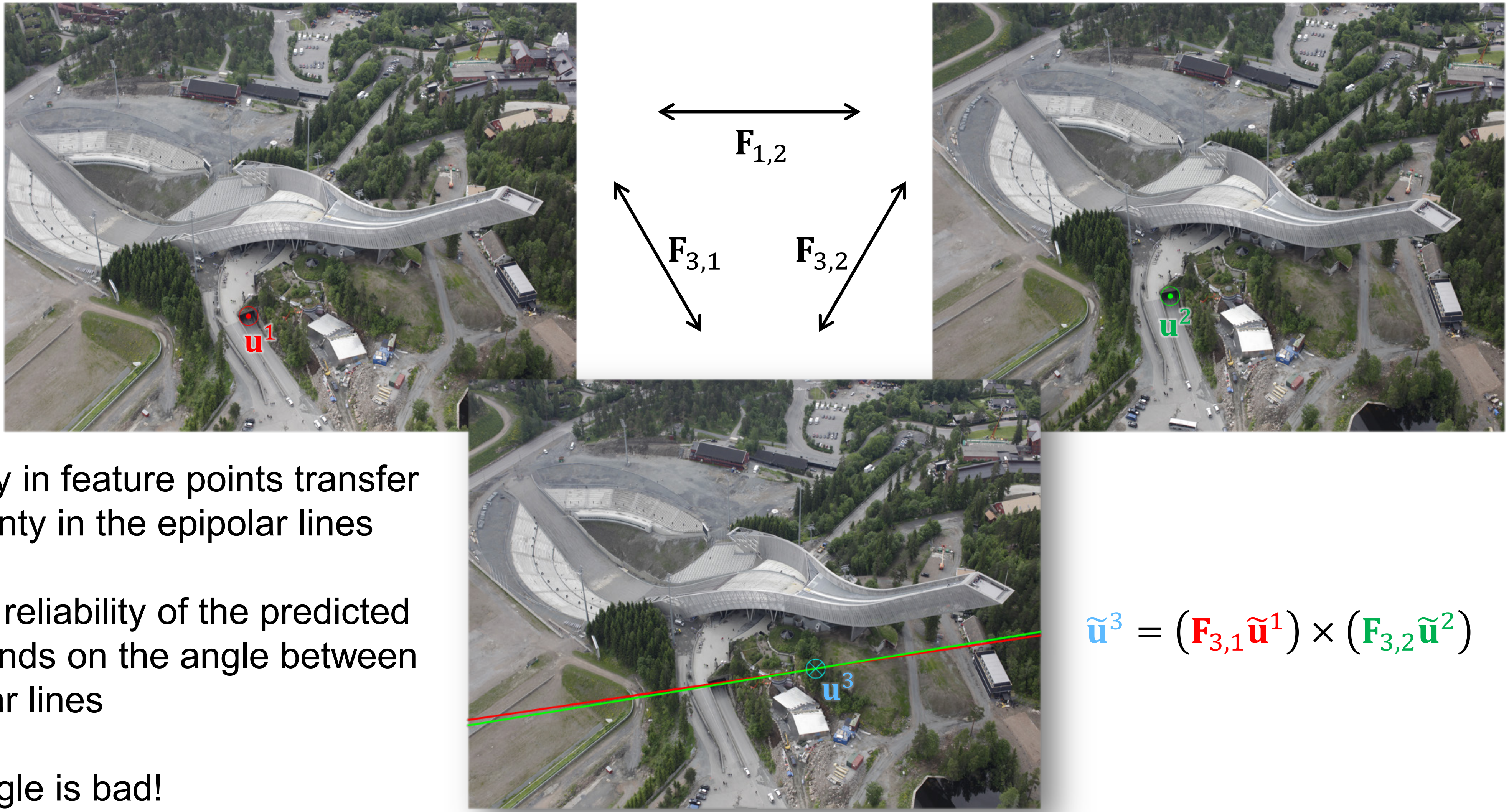
Hence the reliability of the predicted point depends on the angle between the epipolar lines

A large angle is good!

$$\tilde{\mathbf{u}}^3 = (\mathbf{F}_{3,1}\tilde{\mathbf{u}}^1) \times (\mathbf{F}_{3,2}\tilde{\mathbf{u}}^2)$$

Example

Point transfer based on epipolar constraints



$$\tilde{\mathbf{u}}^3 = \left(\mathbf{F}_{3,1}\tilde{\mathbf{u}}^1\right) \times \left(\mathbf{F}_{3,2}\tilde{\mathbf{u}}^2\right)$$

Uncertainty in feature points transfer to uncertainty in the epipolar lines

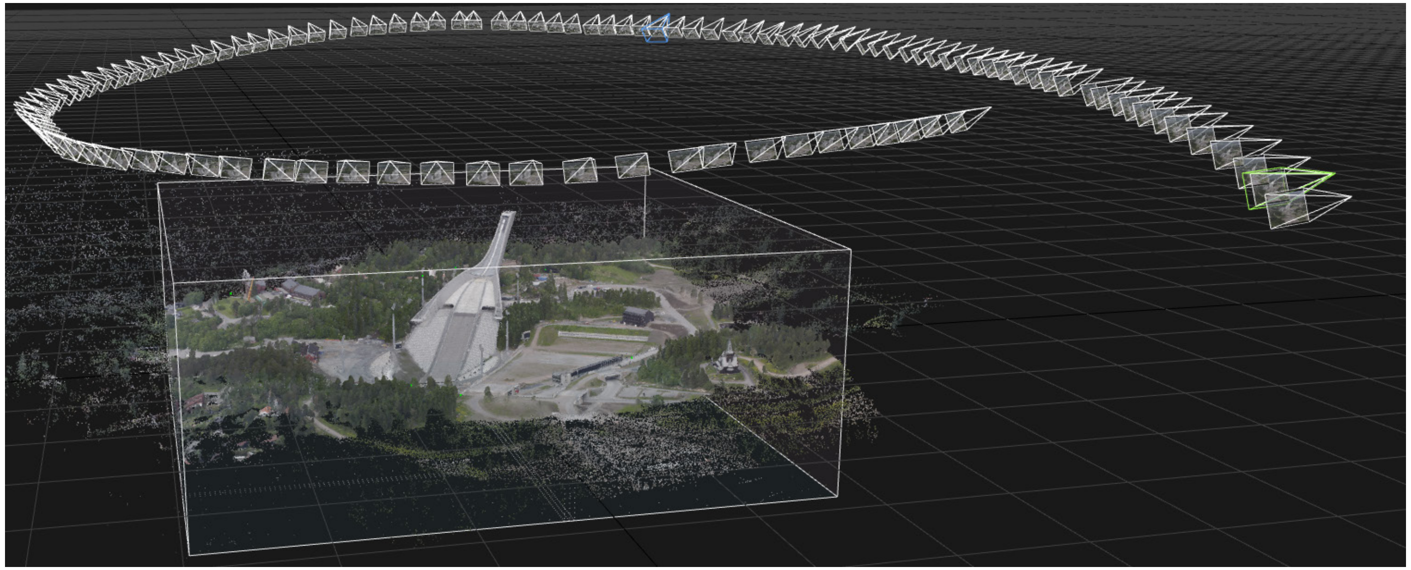
Hence the reliability of the predicted point depends on the angle between the epipolar lines

A small angle is bad!

Multiple-view geometry
Multiple-view stereo

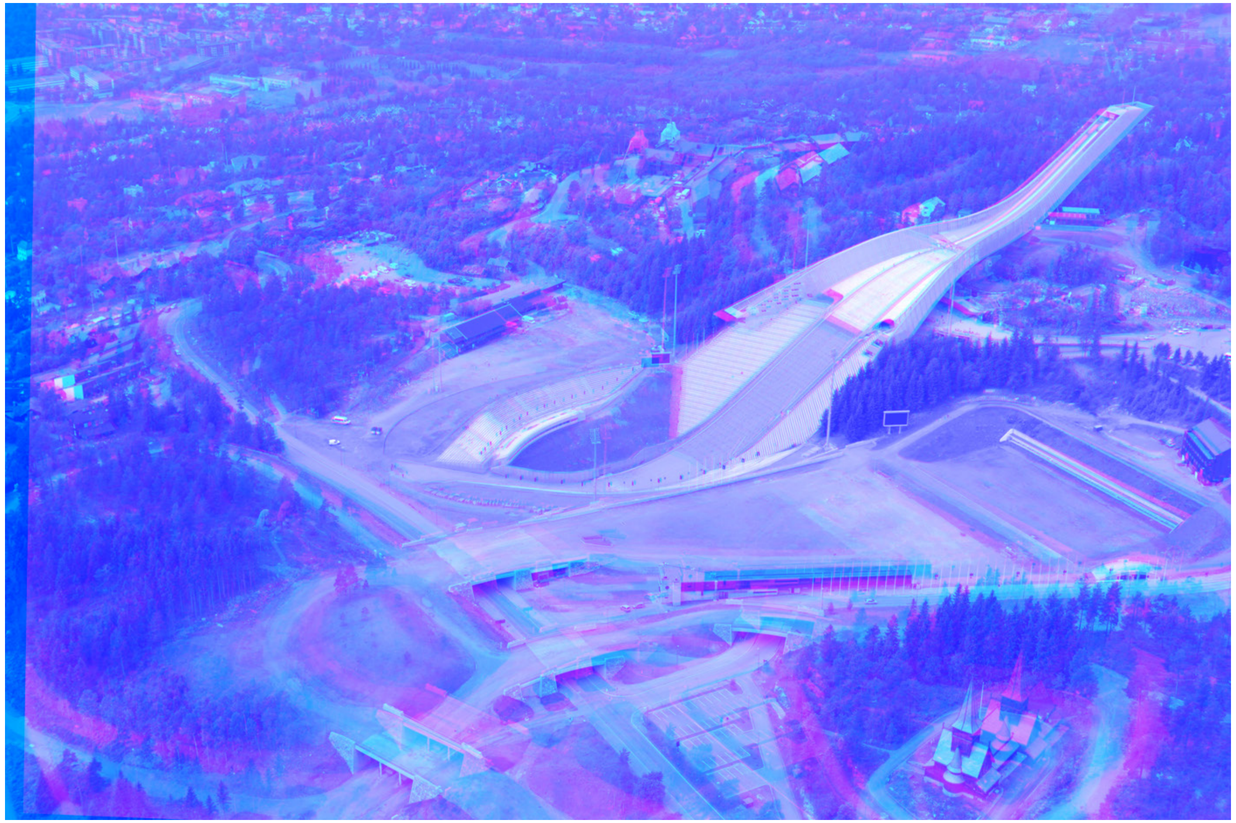
- Multi-view stereo
  - Plane-sweep
  - Volumetric stereo
  - Surface expansion

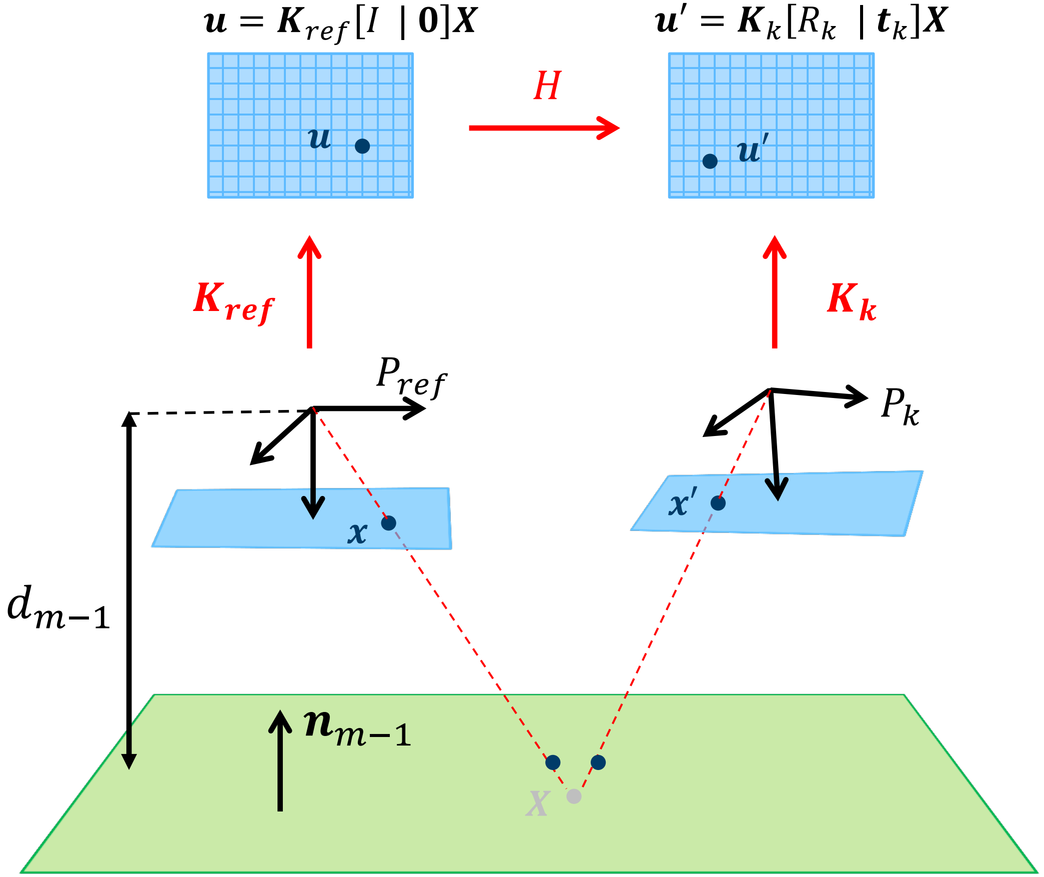
- Surface reconstruction



Plane sweep

- Sweep planes at different depths

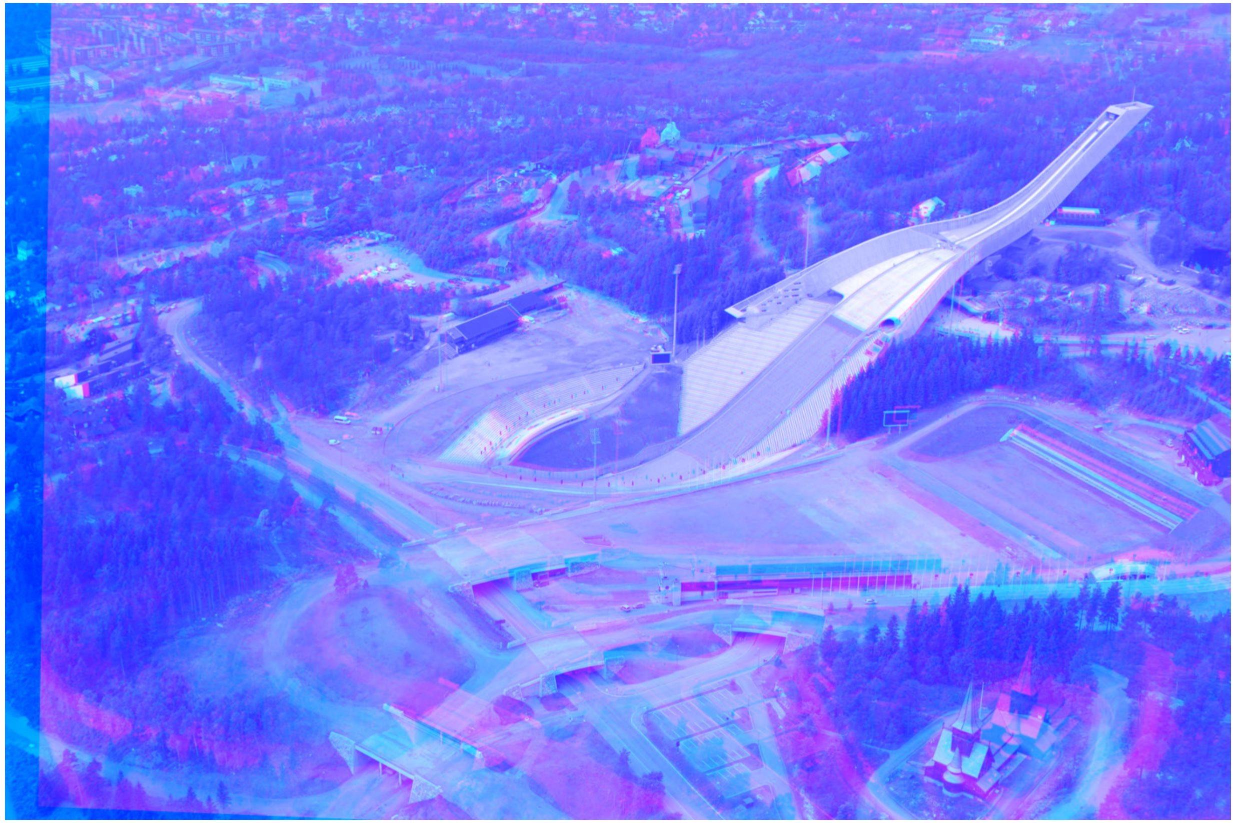


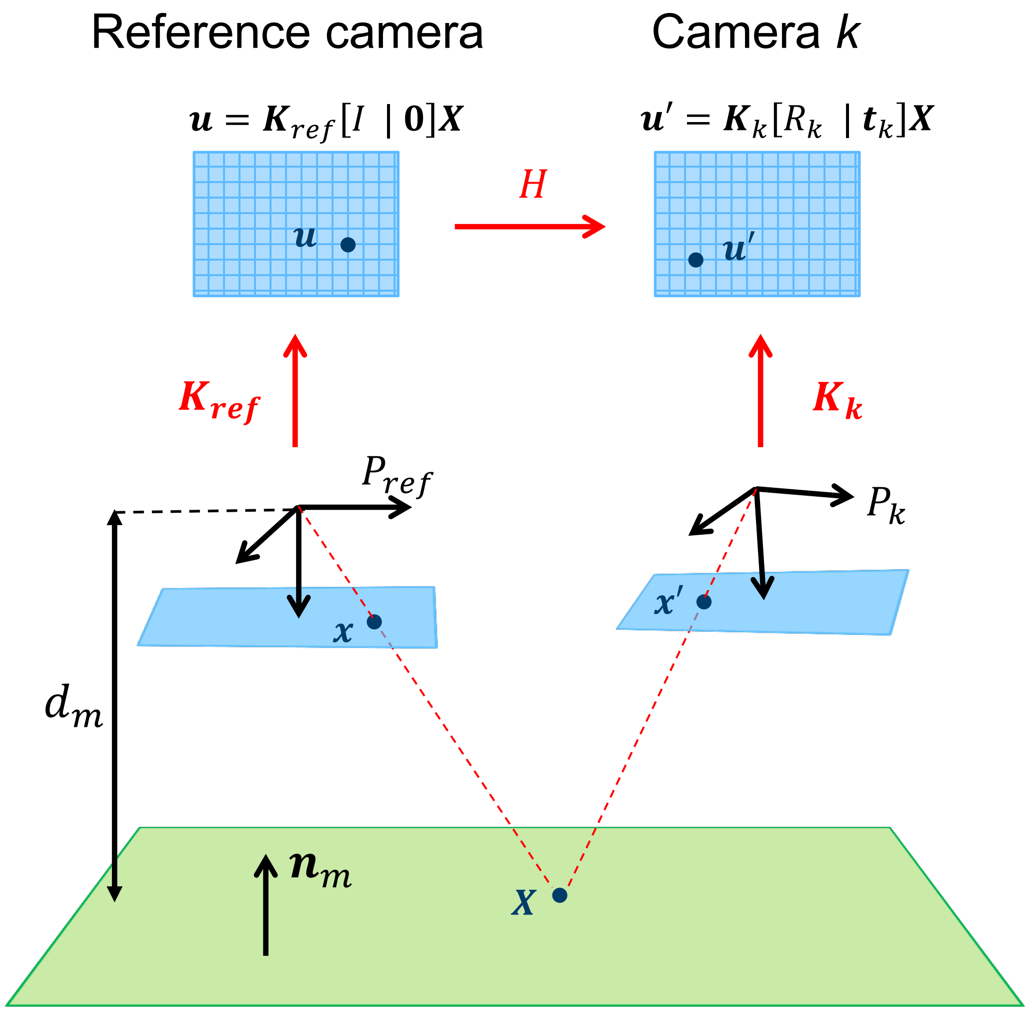


Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

Plane sweep

- Sweep planes at different depths

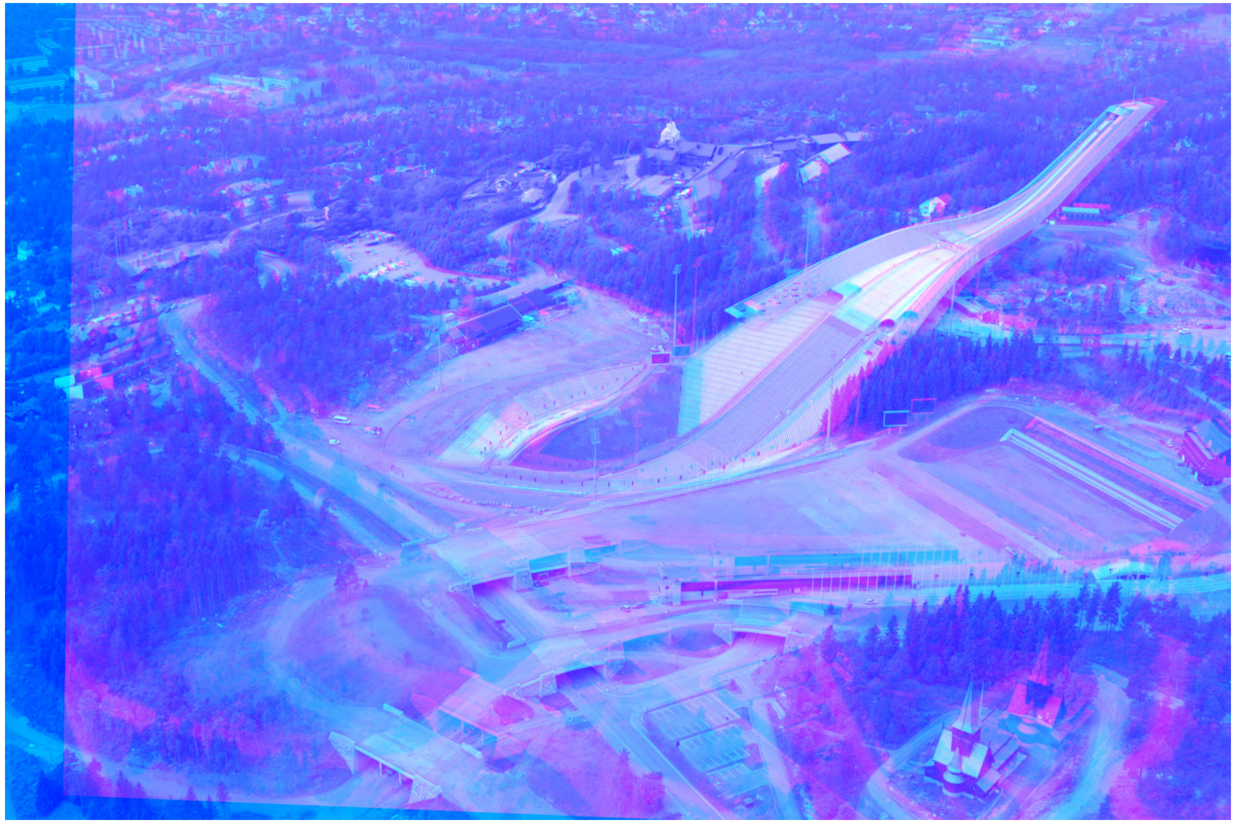


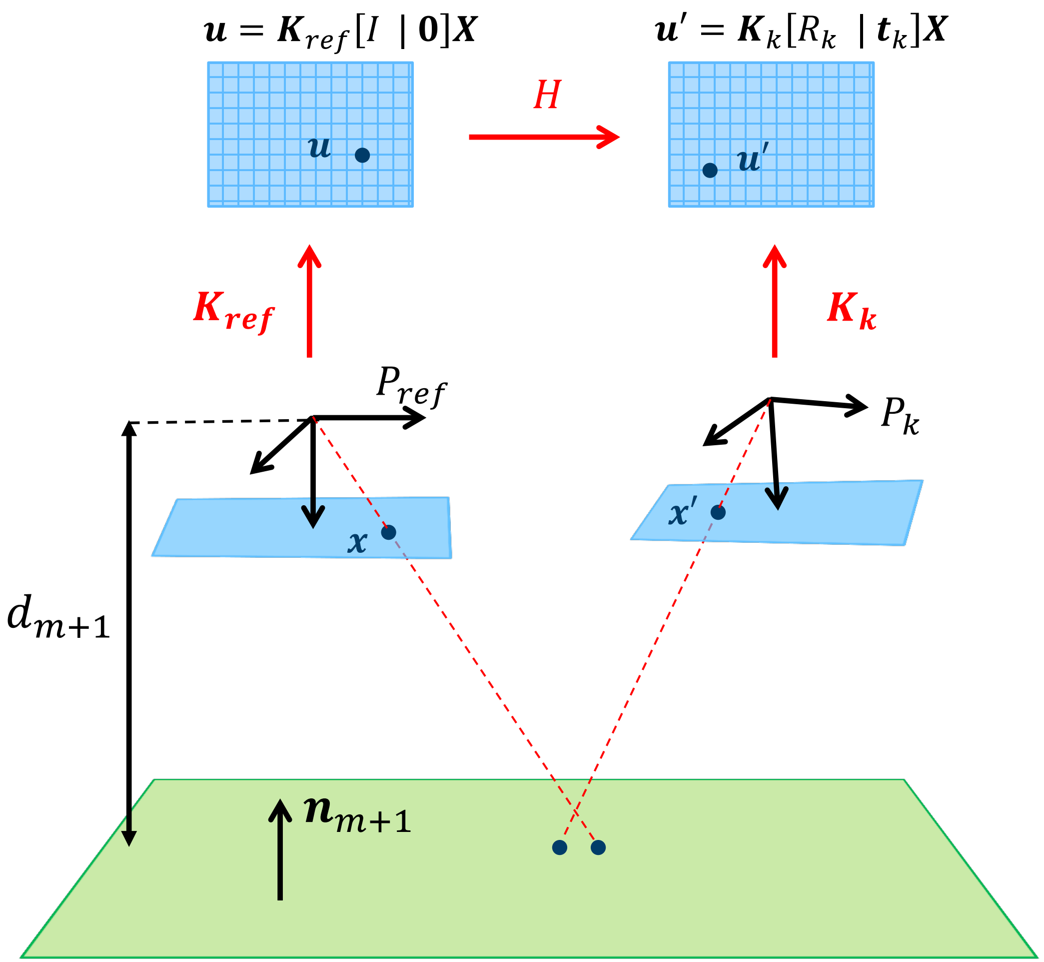


Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

Plane sweep

- Sweep planes at different depths

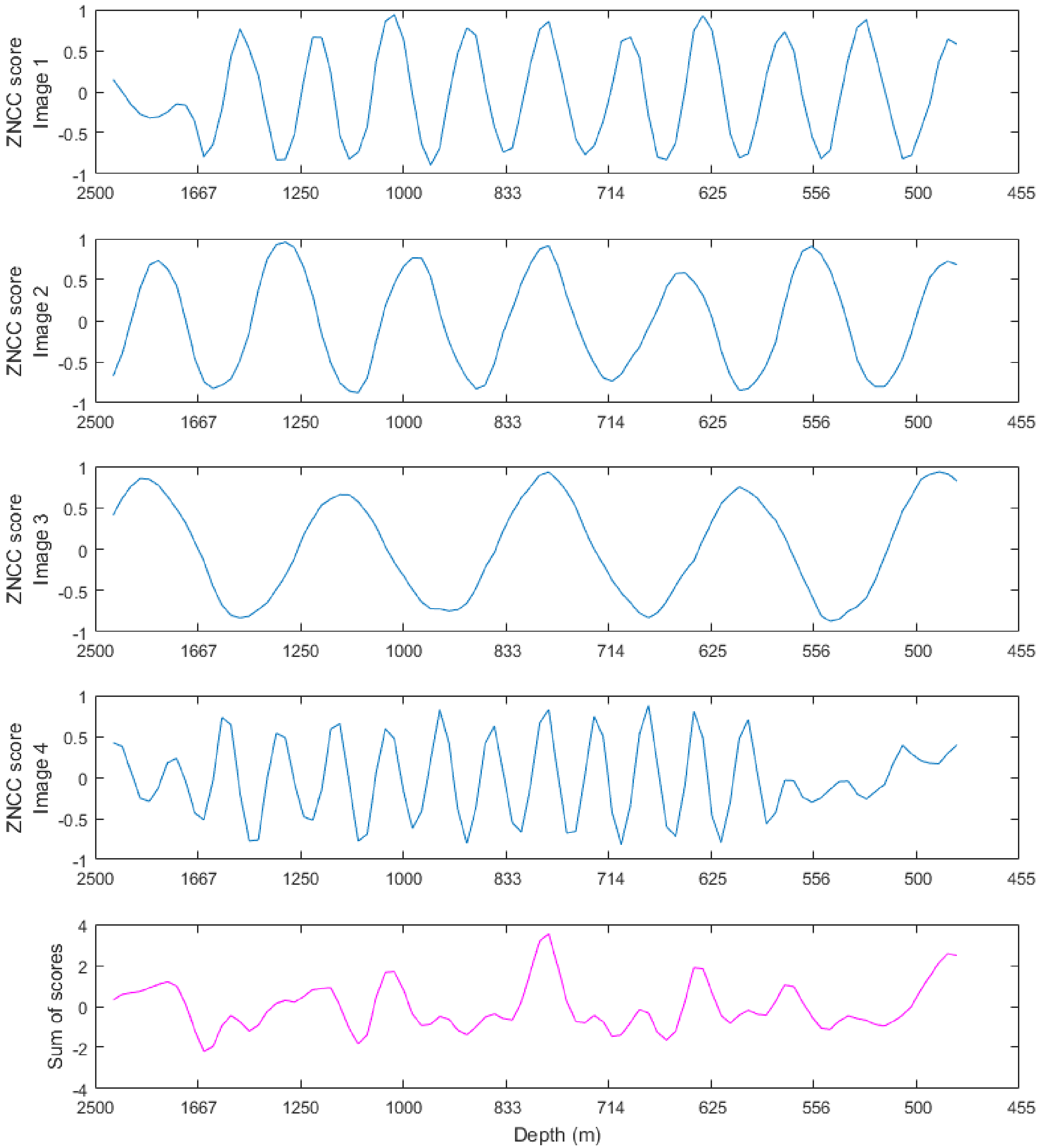




Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

Plane sweep and ambiguities

- Multiple views can resolve ambiguities in difficult areas!



Plane sweep through oriented planes

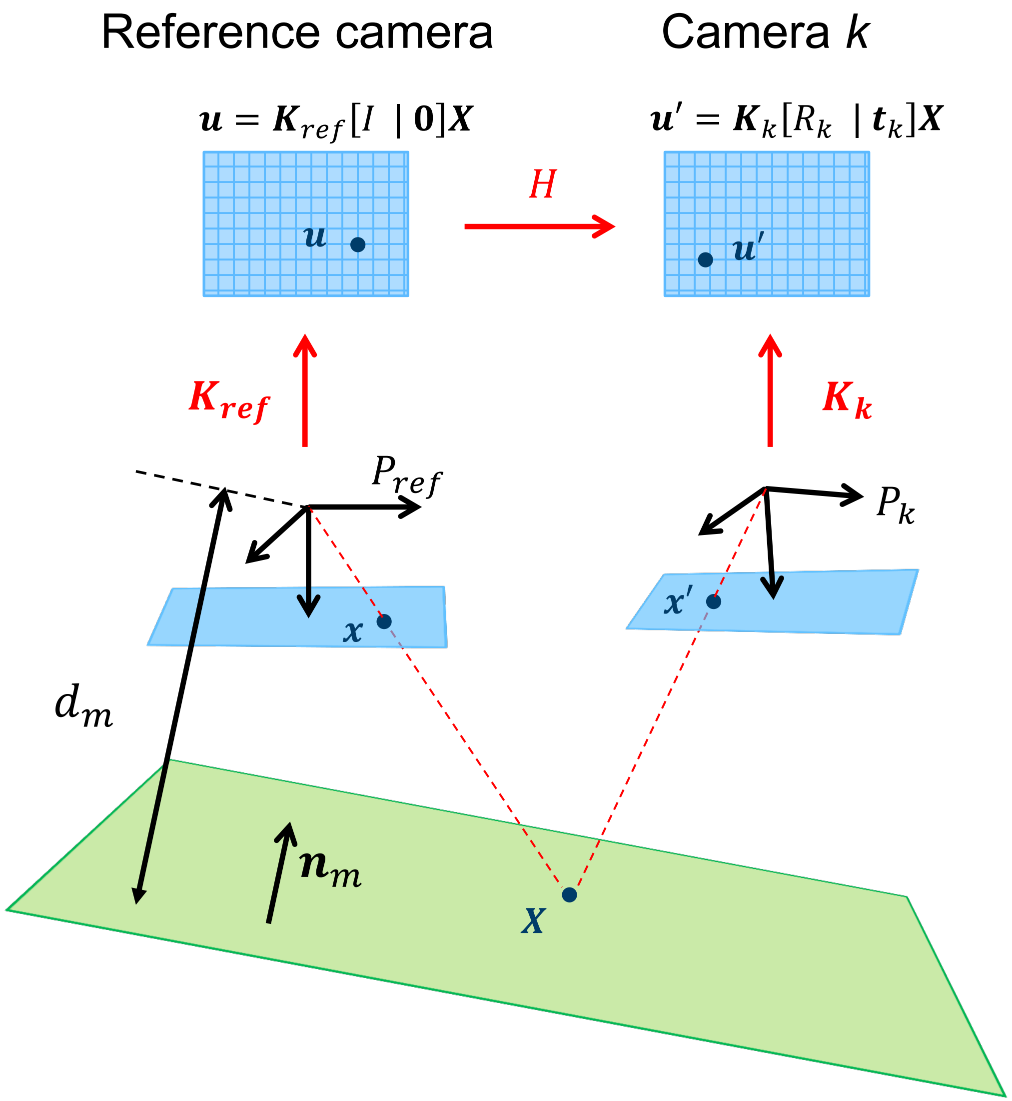
- Fronto-parallel

$$\boldsymbol{n}_m = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$$

$$Z_m(u,v) = d_m$$

- Other plane orientations

$$Z_m(u,v) = \frac{-d_m}{\begin{bmatrix} u & v & 1 \end{bmatrix} K_{ref}^{-T} \boldsymbol{n}_m}$$



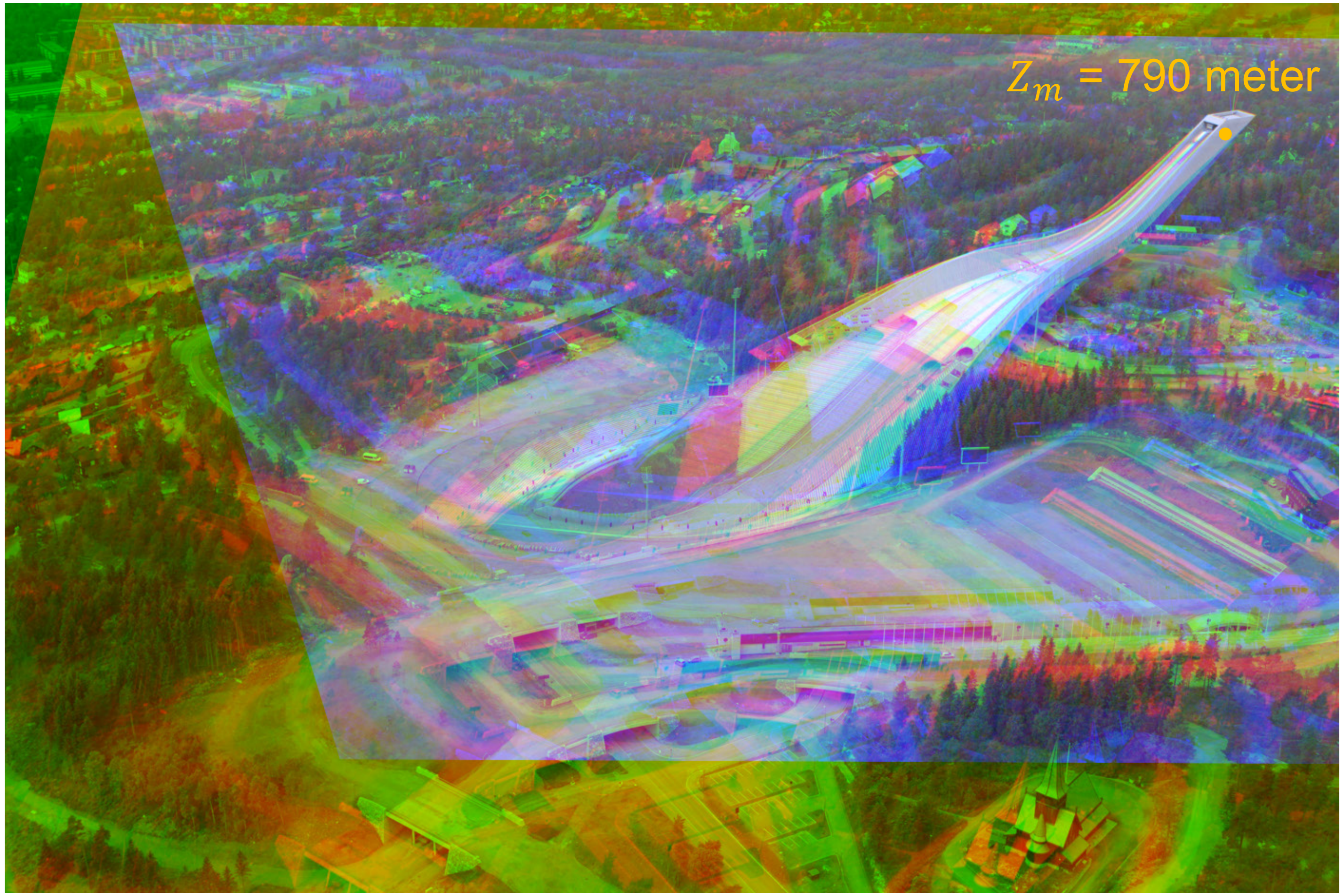
Robert Collins, A Space-Sweep Approach to True Multi-Image Matching, CVPR 1996.
D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions, CVPR 2007

Plane sweep with ground normal

Red:

Green:

Blue:



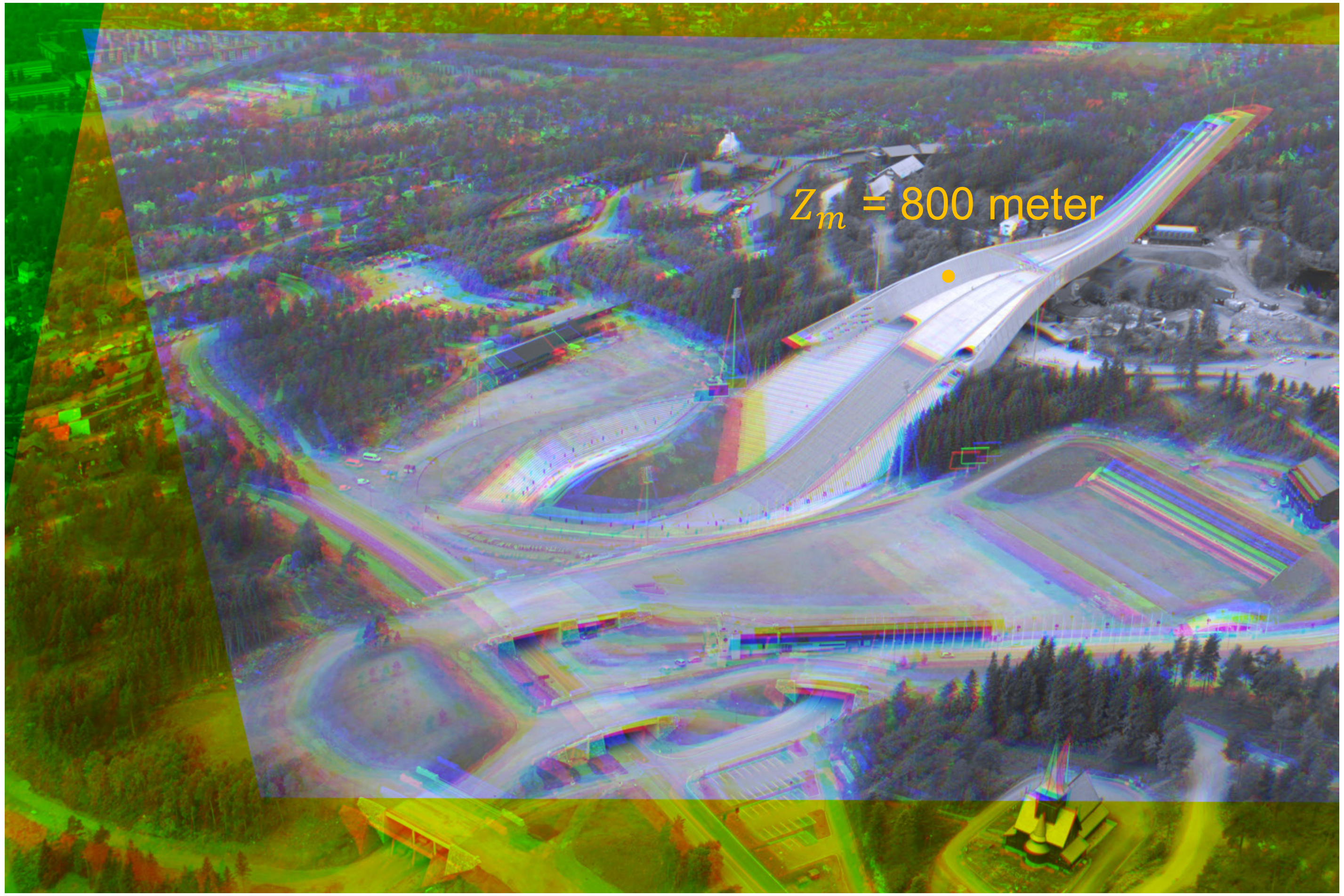
d_m = 200 meter below reference camera

Plane sweep with ground normal

Red:

Green:

Blue:



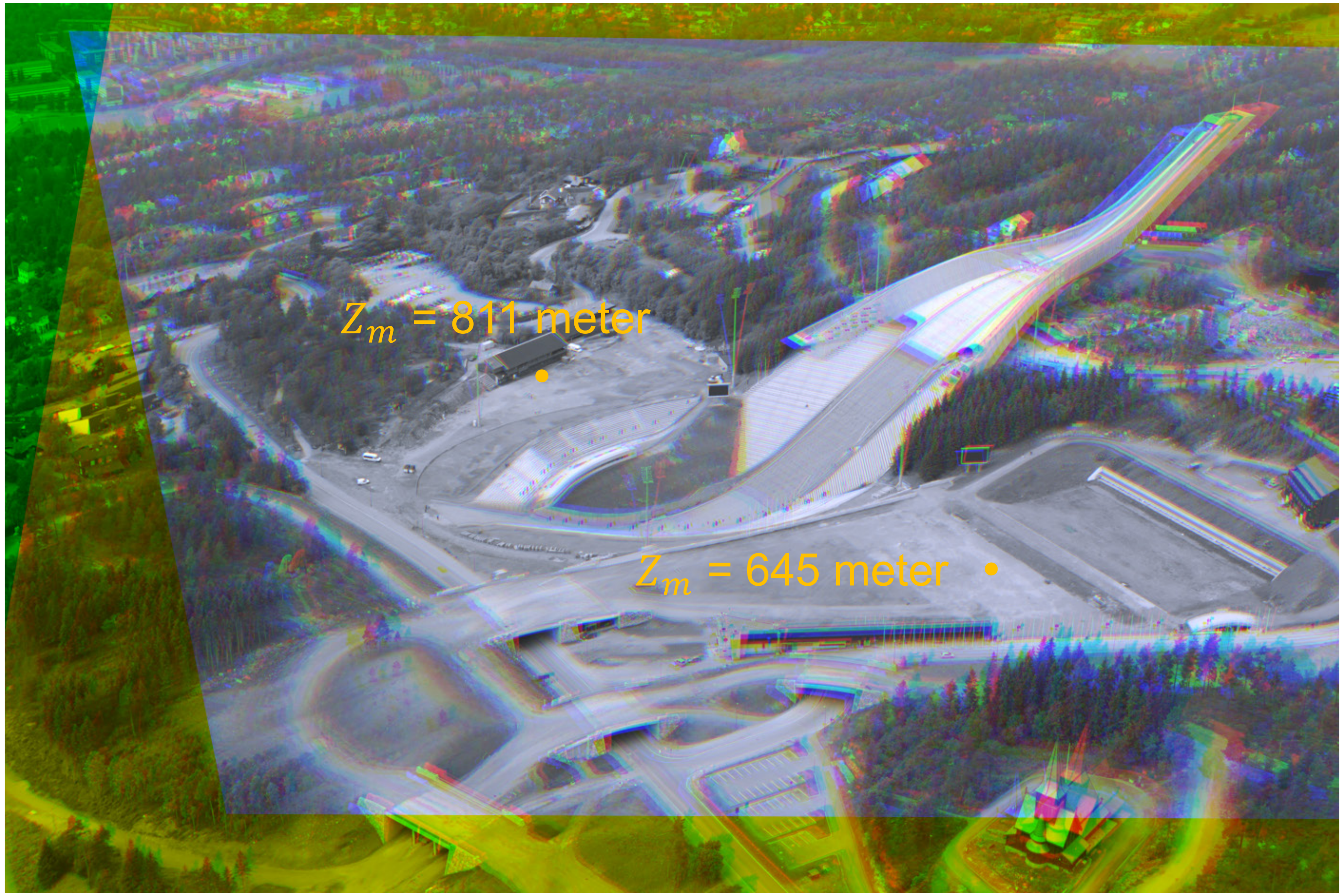
d_m = 261 meter below reference camera

Plane sweep with ground normal

Red:

Green:

Blue:



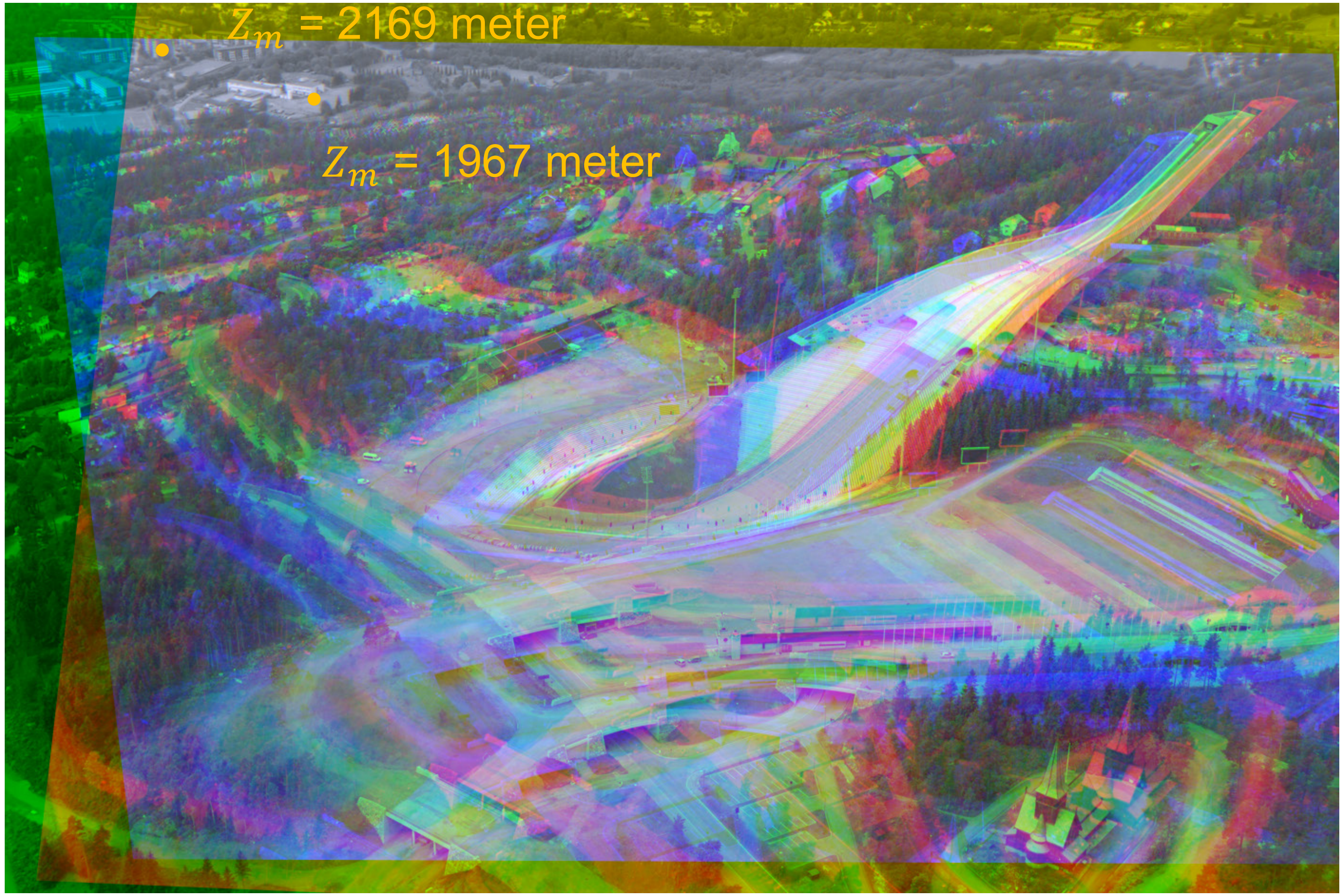
d_m = 298 meter below reference camera

Plane sweep with ground normal

Red:

Green:

Blue:



d_m = 471 meter below reference camera

Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

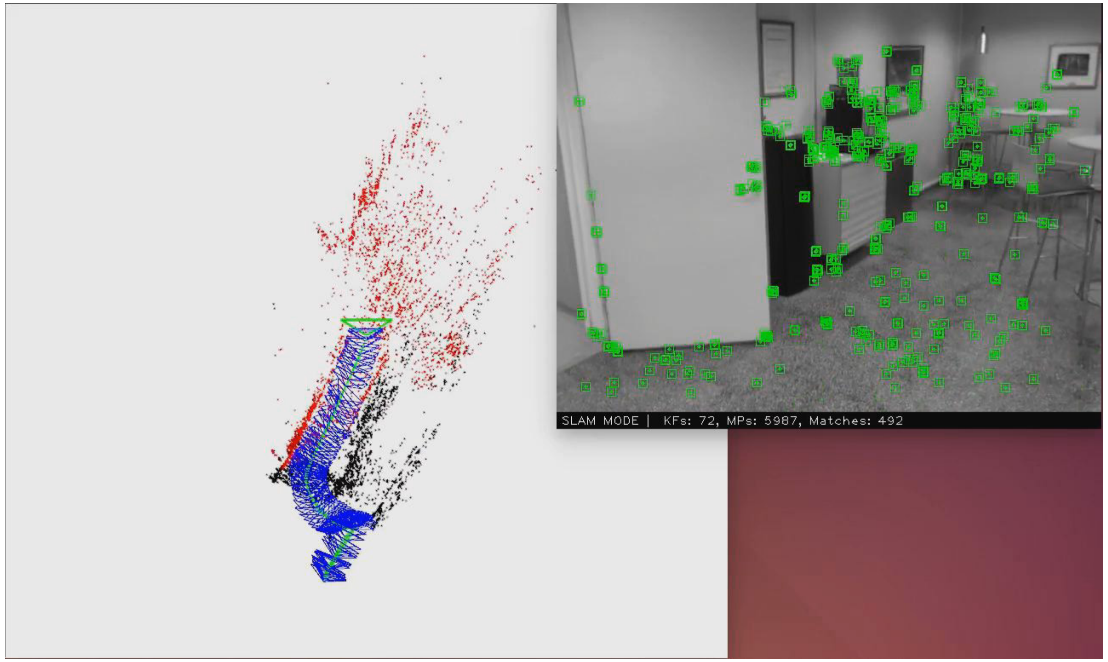
- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

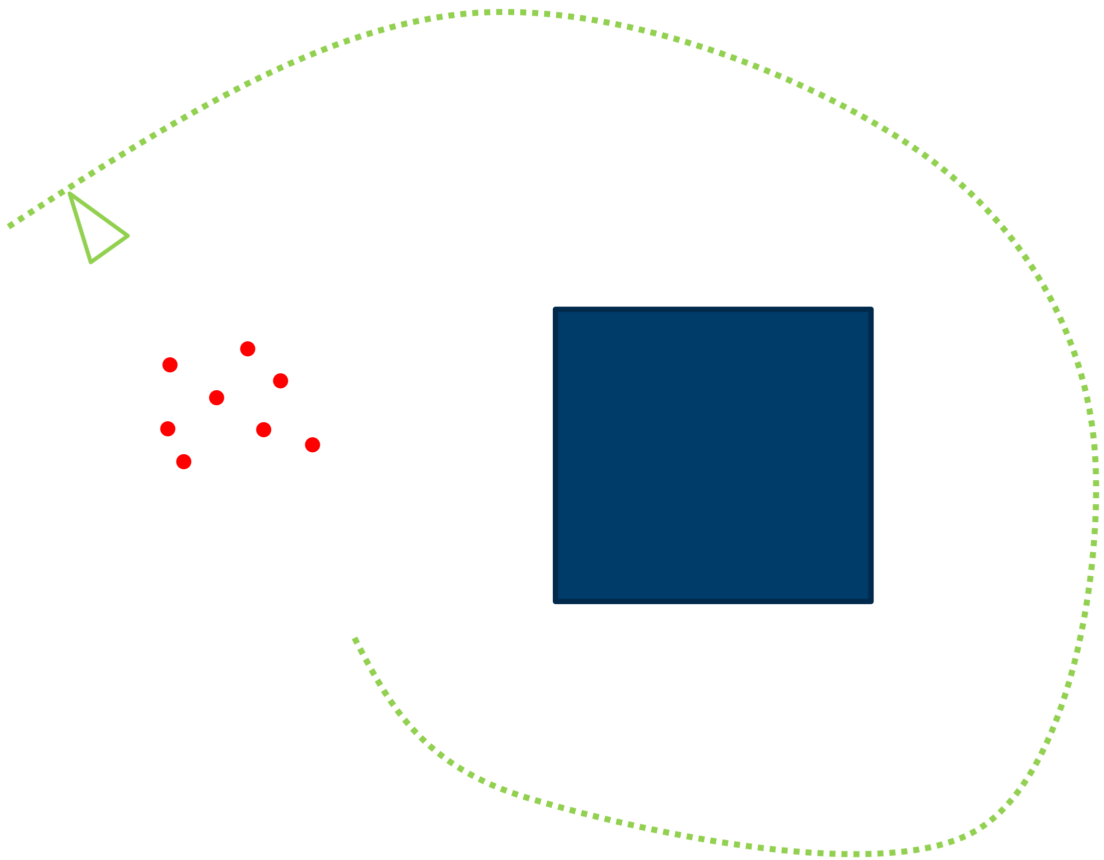
- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

What is Visual SLAM?

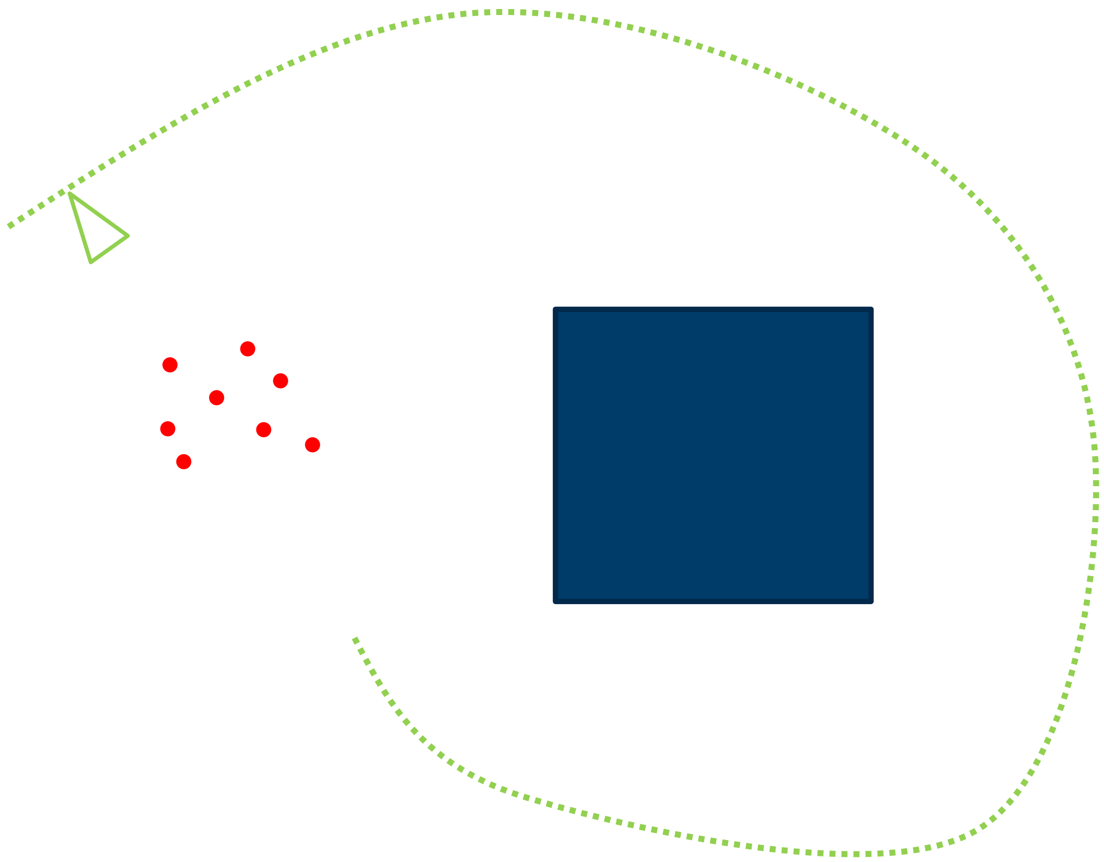
- Visual simultaneous localization and mapping
- Localization (tracking)
  - Localization within the map = tracking the map in image frames
- Mapping
  - Continuously expanding a map while exploring the environment



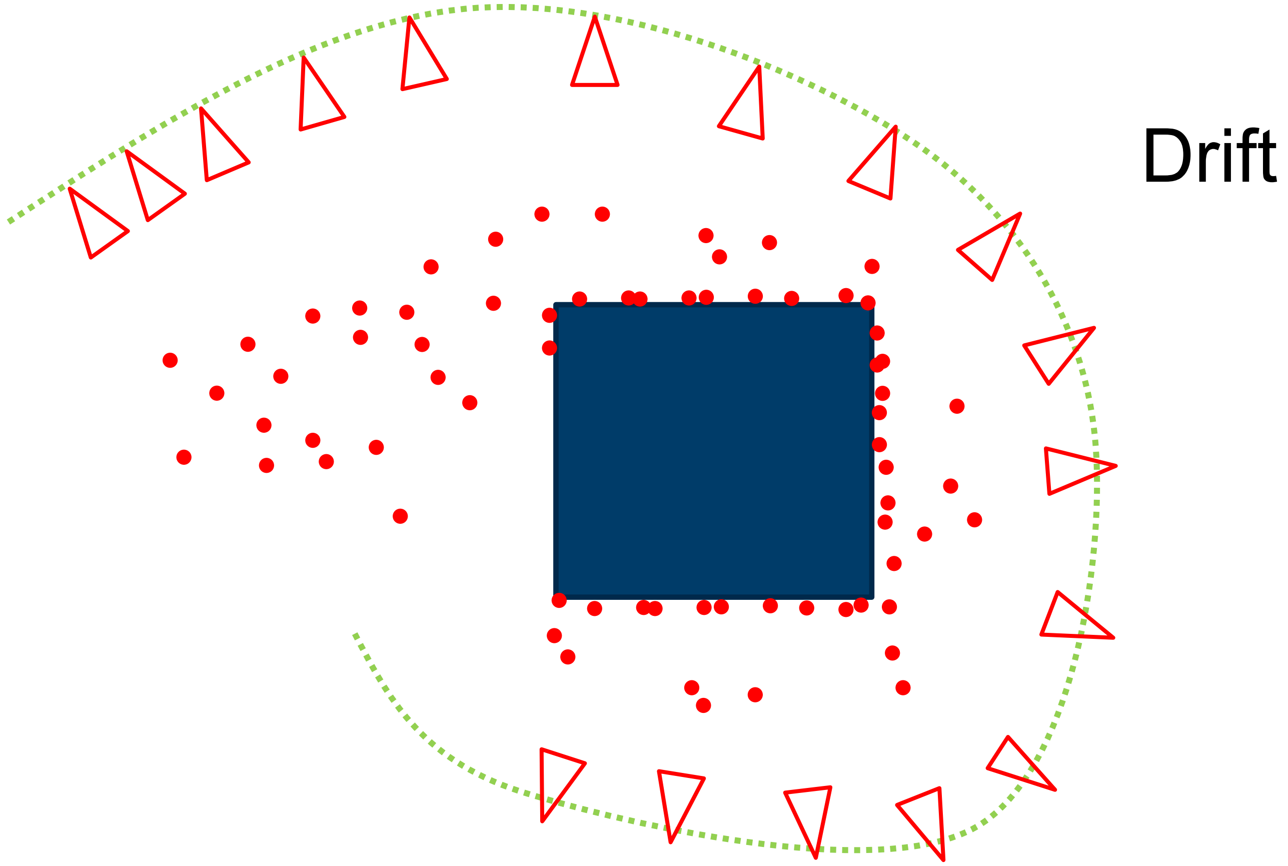
How do we track a map?



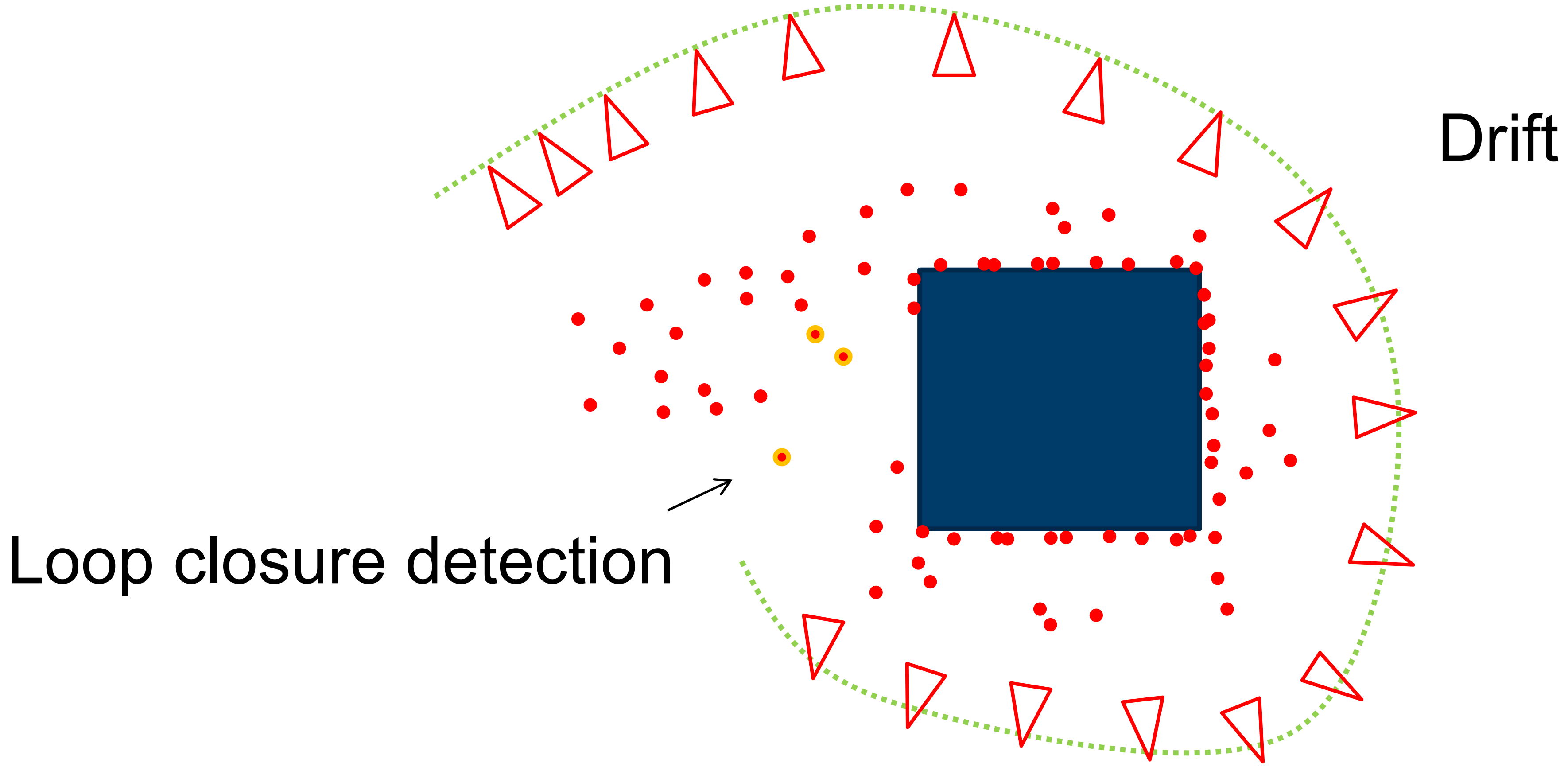
How do we build a map?



Monocular Visual SLAM

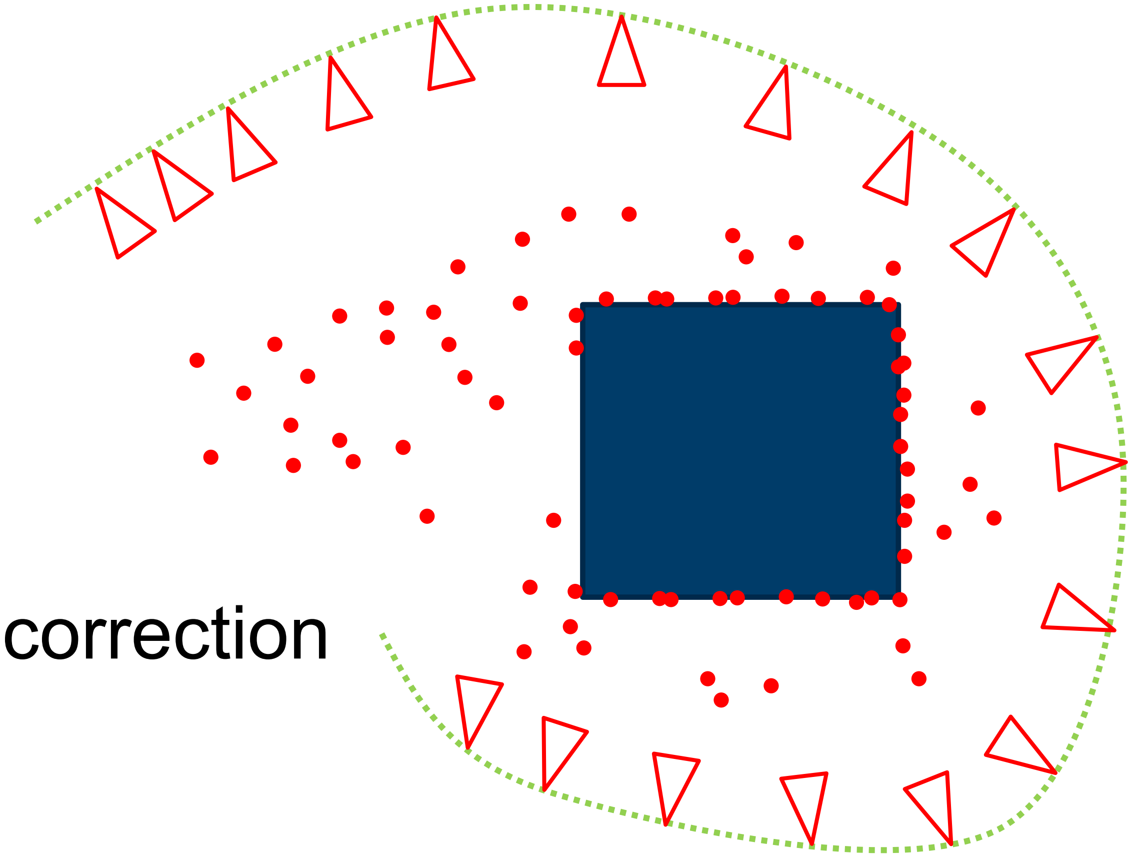


Monocular Visual SLAM

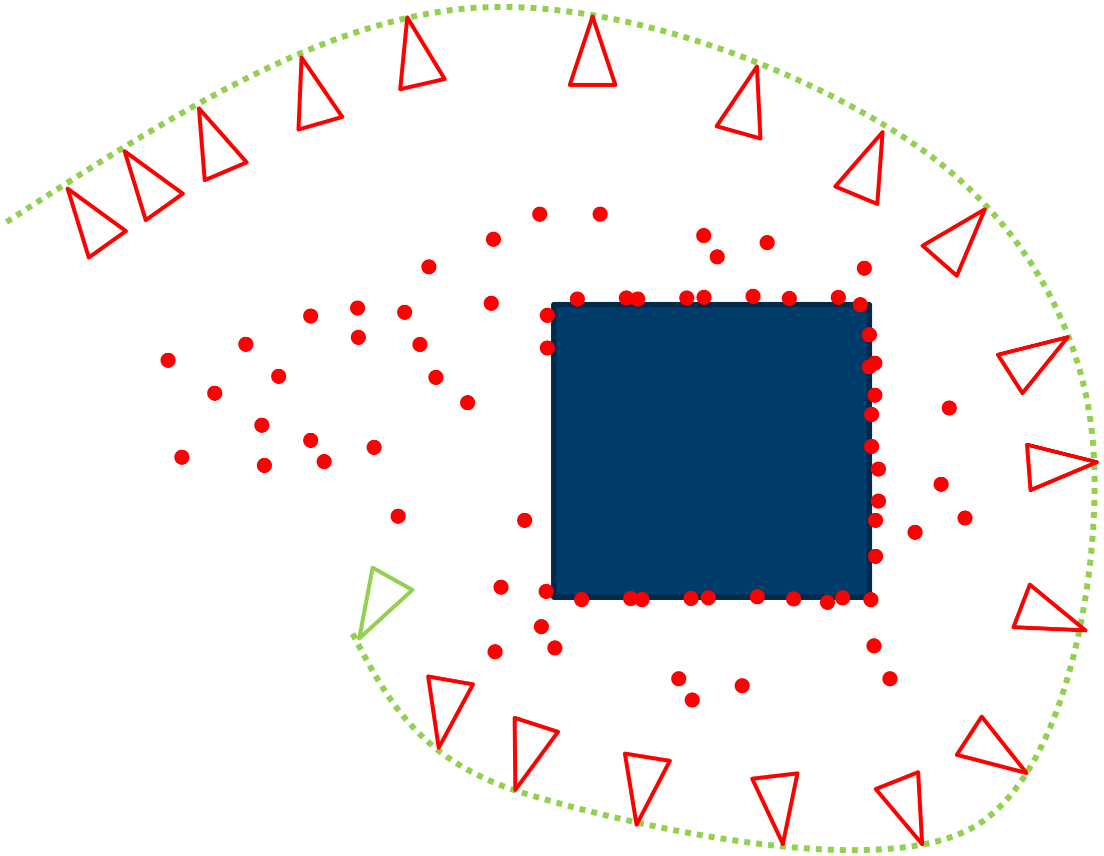


Monocular Visual SLAM

Loop closure correction



Monocular Visual SLAM

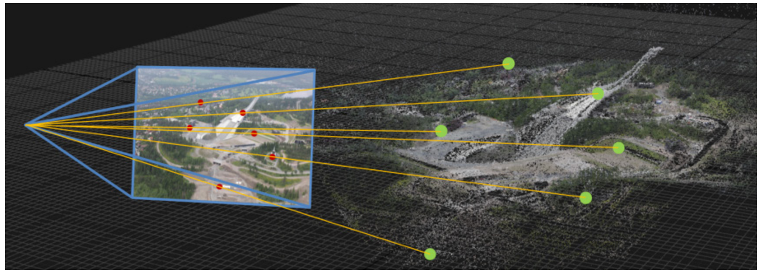


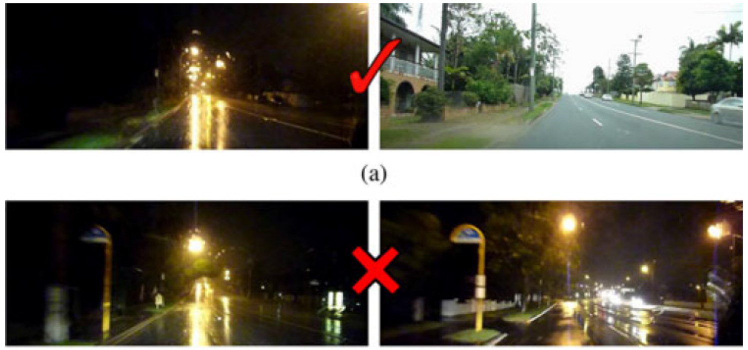
Components of VSLAM

- Short-term tracking
  - Pose estimation given the map
  - Keyframe proposals
- Long-term tracking
  - Visual place recognition
  - Loop closure detection over keyframes
- Mapping
  - Optimizing the map over keyframes
  - Data fusion

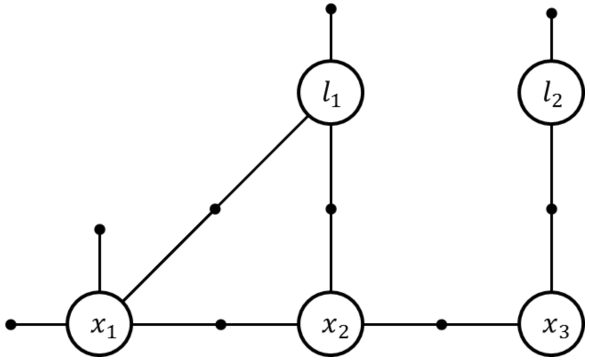
Front end

Back end





Lowry, S. et al. (2016). Visual Place Recognition: A Survey. IEEE Transactions on Robotics, 32(1), 1–19.

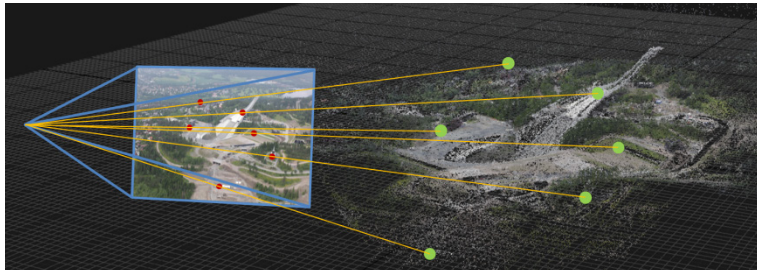


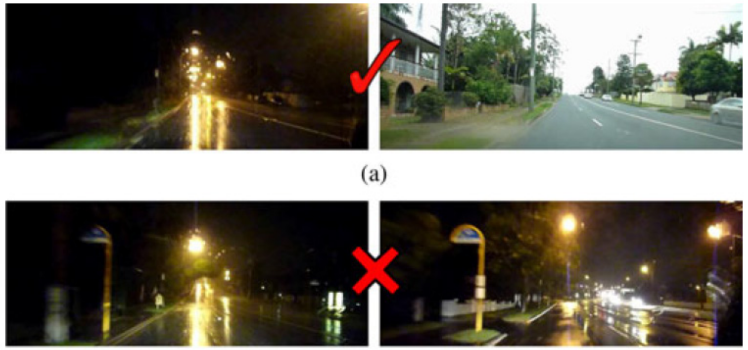
Components of ~~VSLAM~~ VO

- Short-term tracking
  - Pose estimation given the map
  - Keyframe proposals
- ~~Long-term tracking~~
  - ~~Visual place recognition~~
  - ~~Loop closure detection over keyframes~~
- Mapping
  - Optimizing the map over keyframes
  - Data fusion

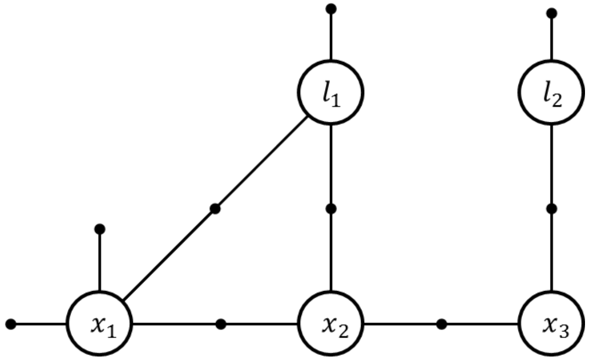
Front end

Back end





Lowry, S. et al. (2016). Visual Place Recognition: A Survey. IEEE Transactions on Robotics, 32(1), 1–19.

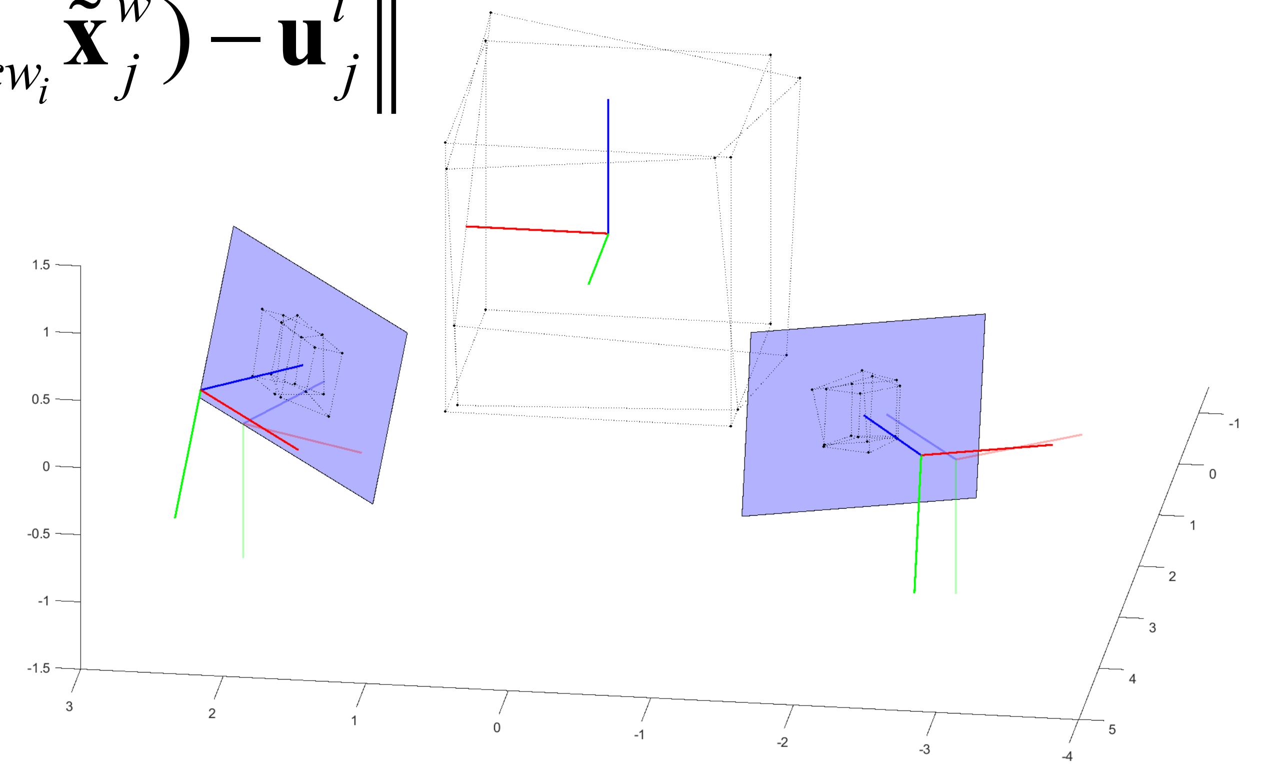


Pose and structure estimation by minimizing reprojection error

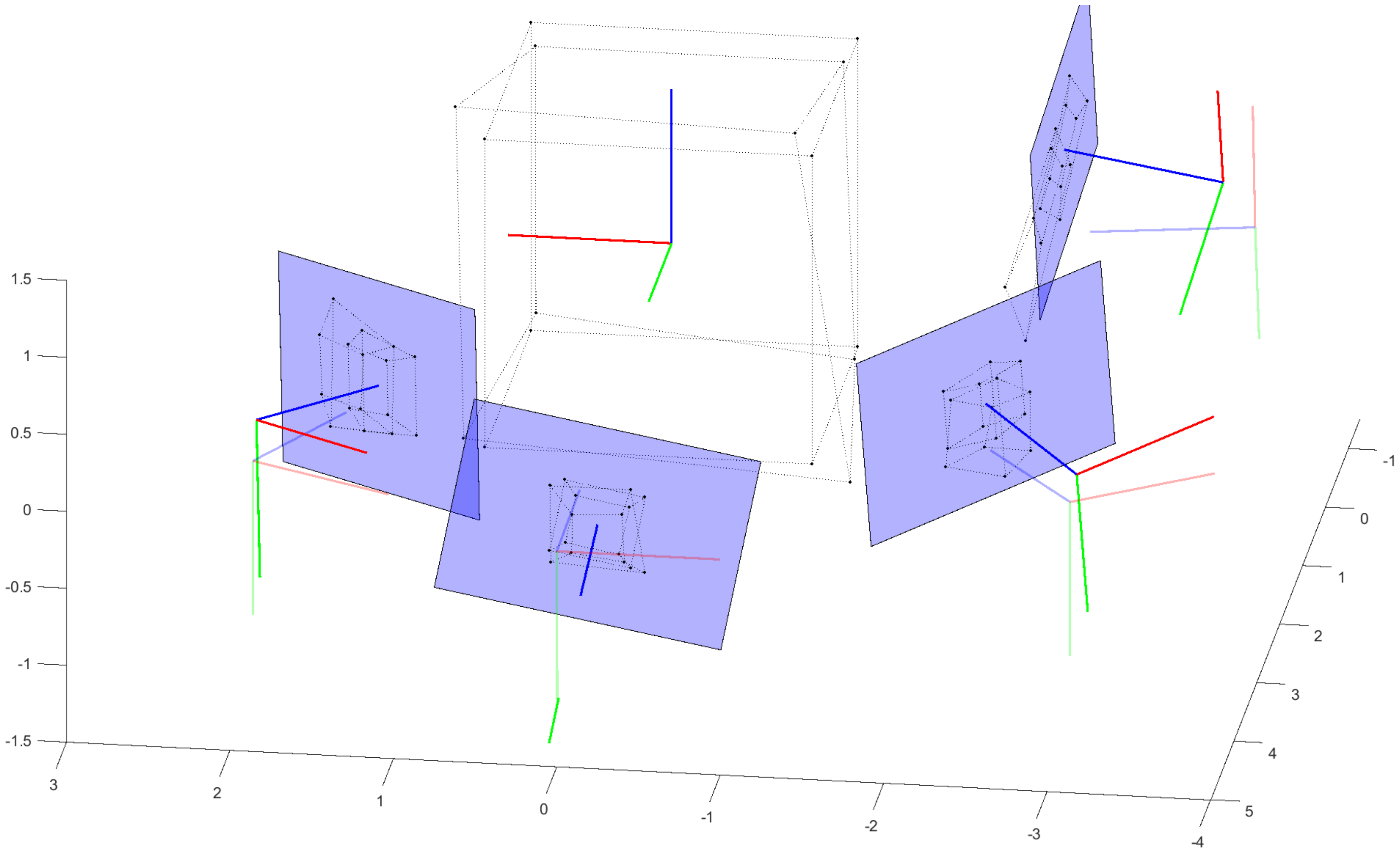
Minimize **geometric error** over the **camera poses** and **world points**

This is also sometimes called **Full Bundle Adjustment**

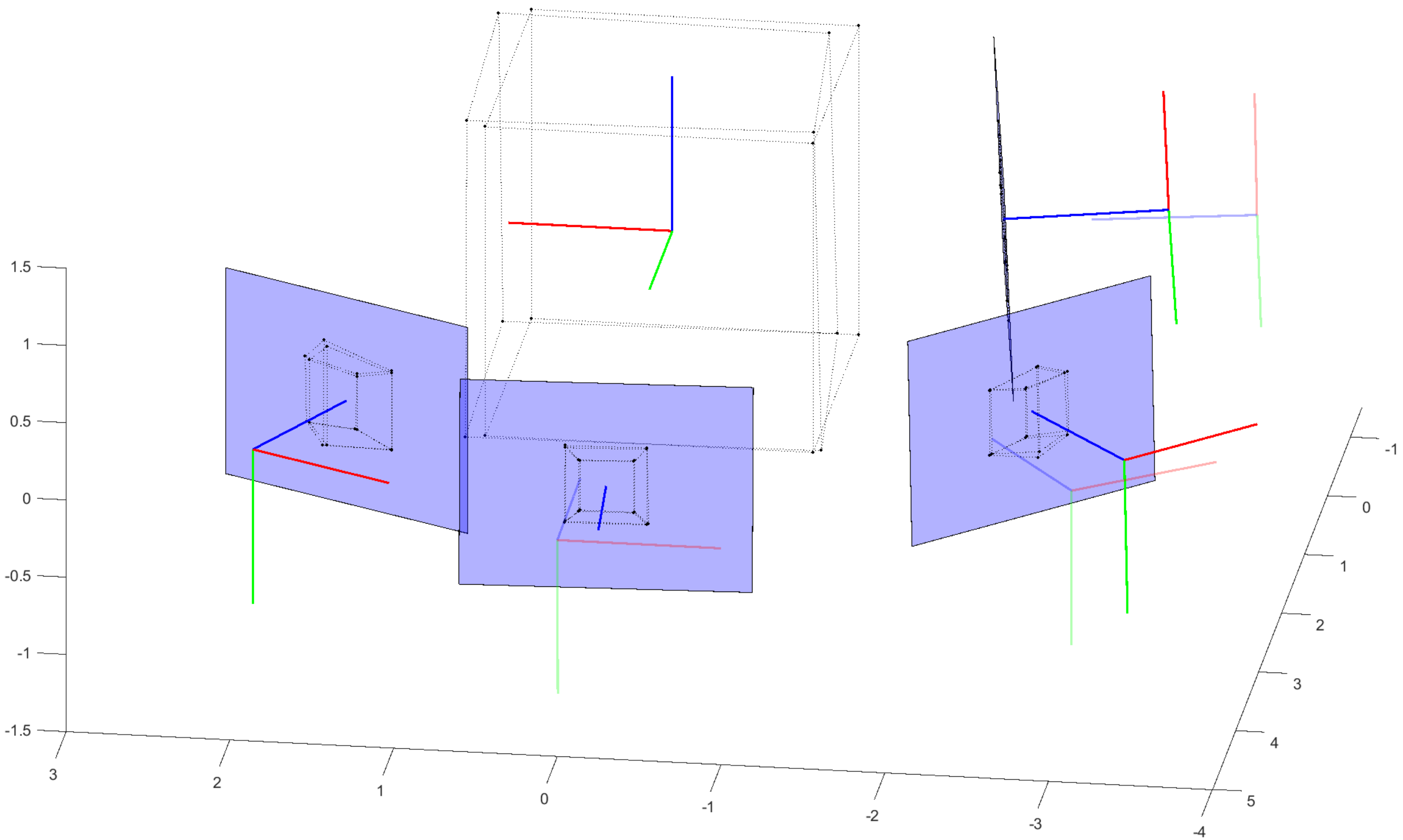
$$\left\{\mathbf{T}_{cw_i}^*, \mathbf{x}_j^{w*}\right\} = \underset{\mathbf{T}_{cw_i}, \mathbf{x}_j^w}{\operatorname{argmin}} \sum_i \sum_j \left\| \pi_i(\mathbf{T}_{cw_i} \tilde{\mathbf{x}}_j^w) - \mathbf{u}_j^i \right\|^2$$



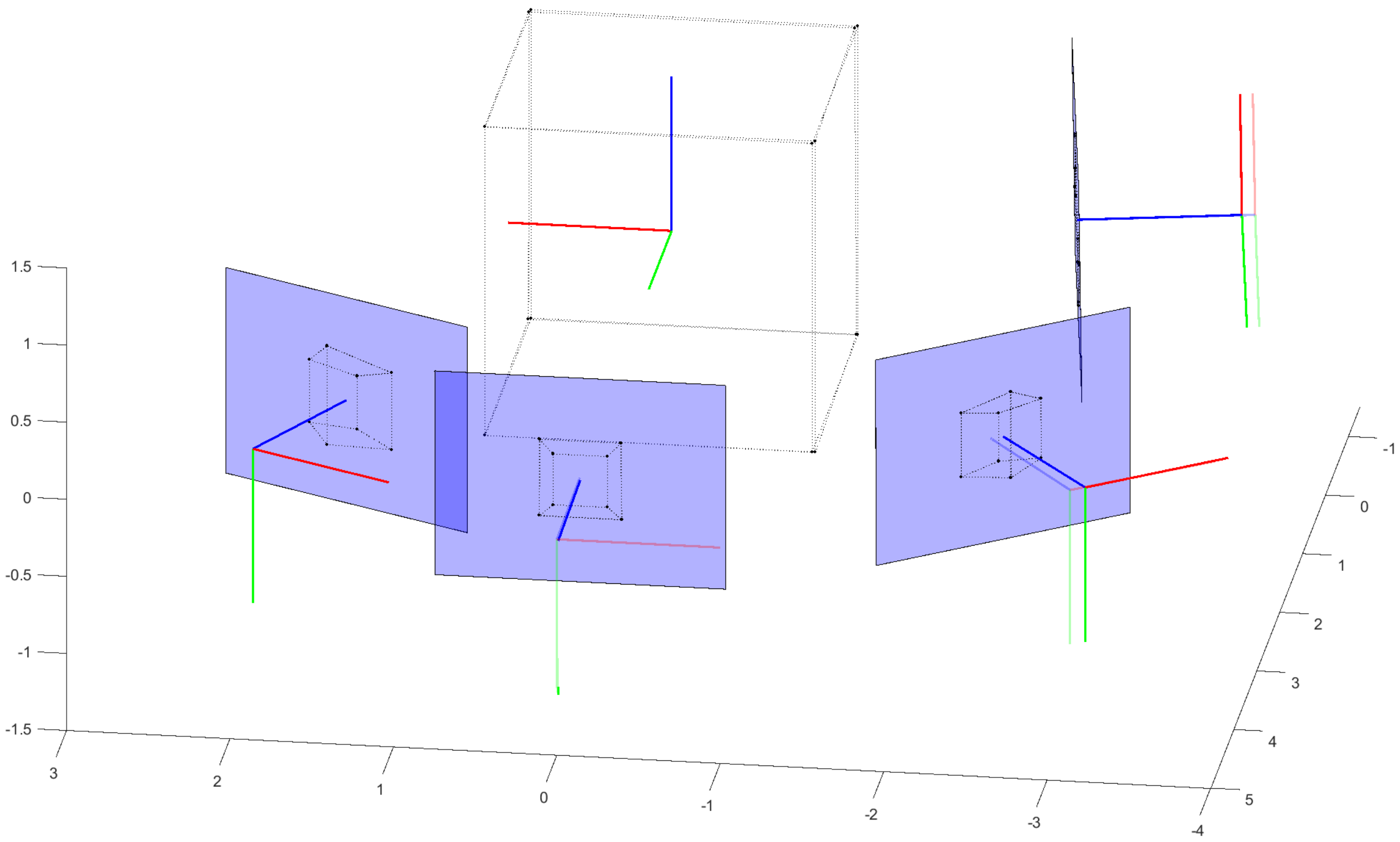
Example



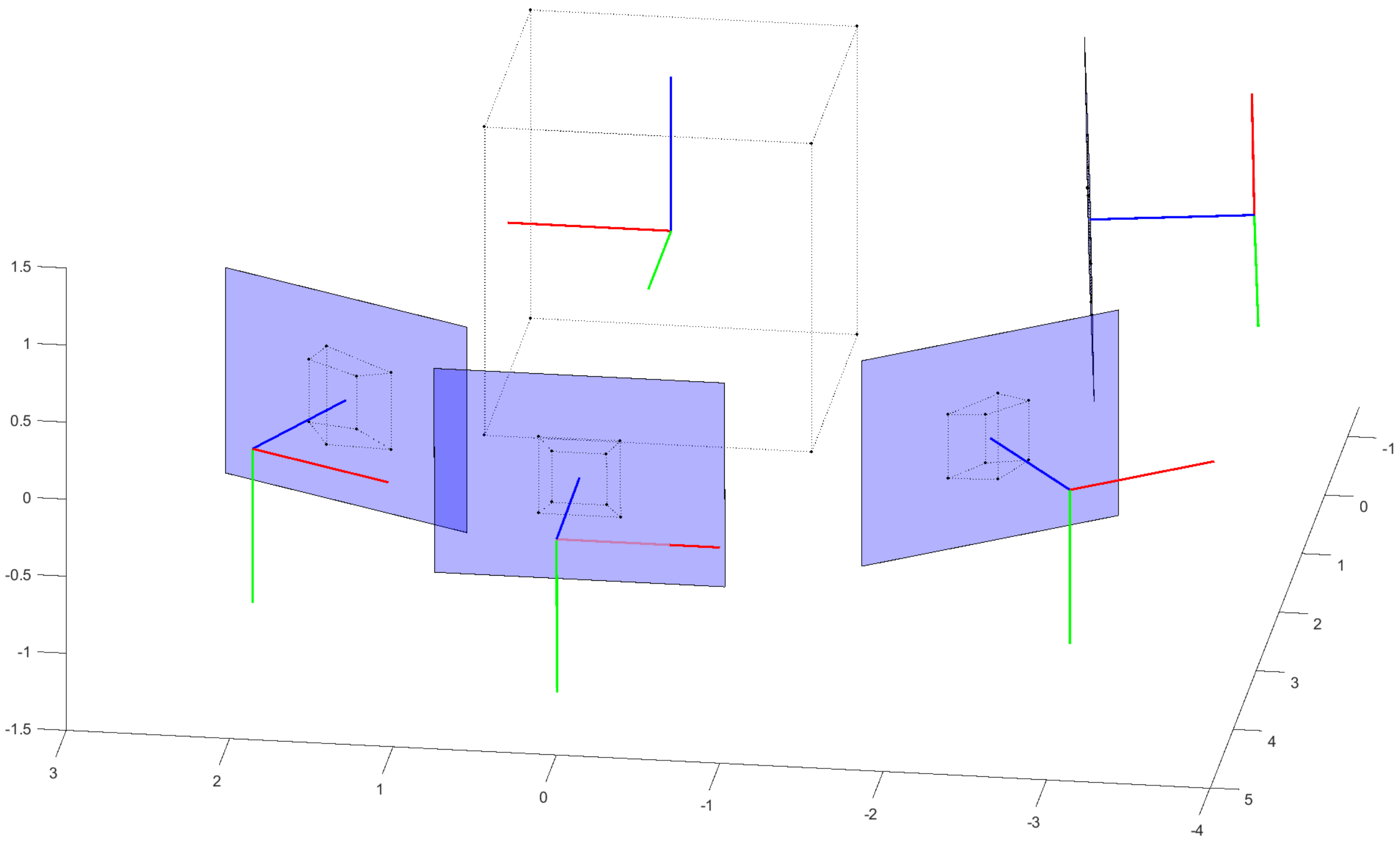
Example



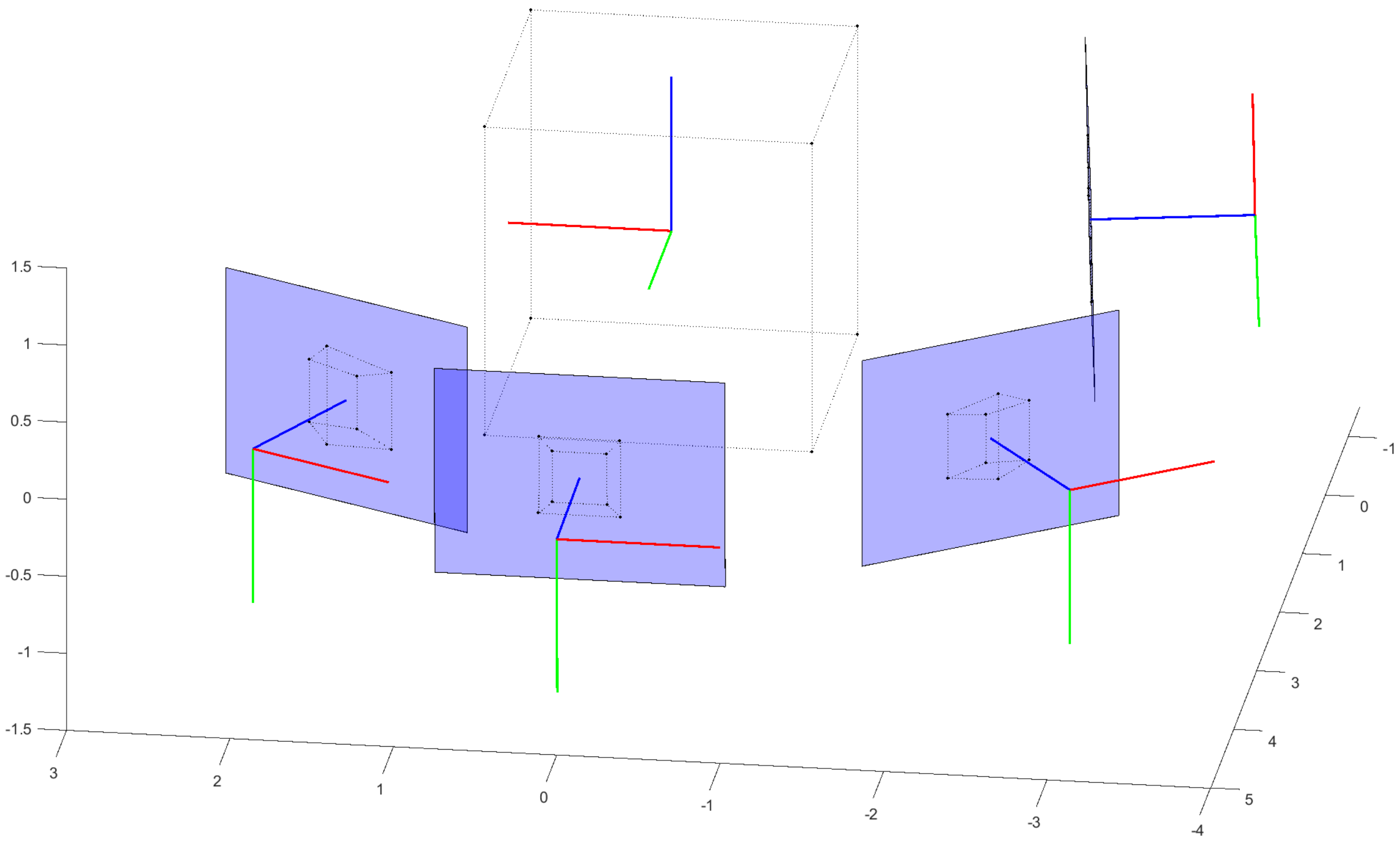
Example



Example



Example



Linearized least-squares

Prior on first pose and distance between first two points

$$\mathbf{A} = \begin{bmatrix} \mathbf{F}_{11} & & \mathbf{G}_{11} & & \\ \mathbf{F}_{12} & & & \mathbf{G}_{12} & \\ \mathbf{F}_{13} & & & & \mathbf{G}_{13} \\ & \mathbf{F}_{21} & \mathbf{G}_{21} & & \\ & \mathbf{F}_{22} & & \mathbf{G}_{22} & \\ & \mathbf{F}_{23} & & & \mathbf{G}_{23} \\ \mathbf{I}_{2\times 6} & & & & \\ & & \mathbf{D}_{12} & -\mathbf{D}_{12} & \end{bmatrix} \quad \boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \\ \boldsymbol{\delta}_3 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{12} \\ \mathbf{b}_{13} \\ \mathbf{b}_{21} \\ \mathbf{b}_{22} \\ \mathbf{b}_{23} \\ \mathbf{b}_{\xi_1^{prior}} \\ \mathbf{b}_{d_{12}^{prior}} \end{bmatrix}$$

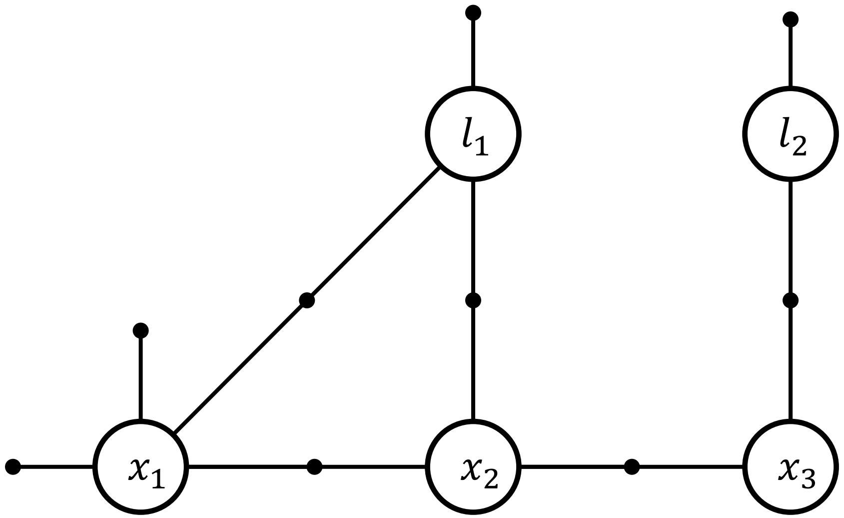
$$\mathbf{b}_{\xi_1^{prior}} = \ln\left(\mathbf{T}_{wc_1}^{-1} \mathbf{T}_{wc_1}^{prior}\right)^{\vee}$$

$$\mathbf{b}_{d_{12}^{prior}} = d_{12}^{prior} - \left\| \mathbf{x}_2^w - \mathbf{x}_1^w \right\|$$

MAP inference for nonlinear factor graphs

MAP inference for factor graphs:

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\, \phi(X) = \underset{X}{\operatorname{argmax}} \prod_i \phi_i(X_i)$$



Let us assume that all factors are of the form

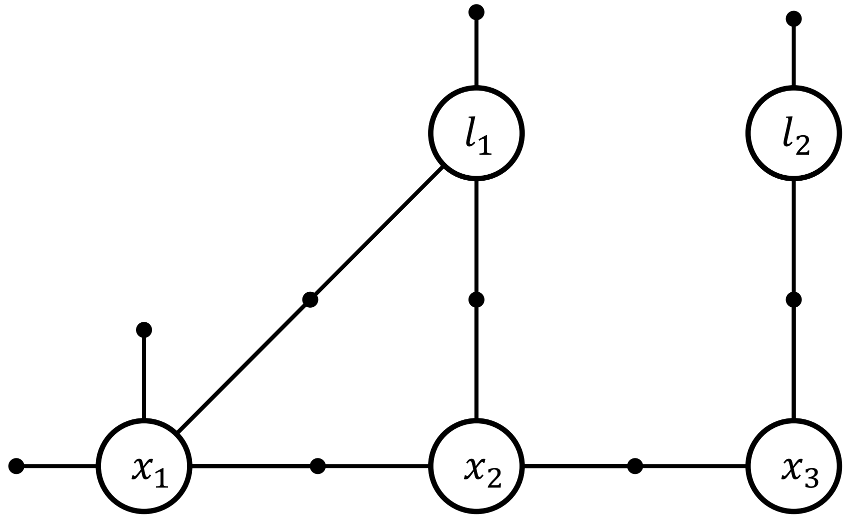
$$\phi_i(X_i) \propto \exp\left\{-\frac{1}{2}\left\|h_i(X_i) - z_i\right\|^2_{\Sigma_i}\right\}$$

Taking the negative log and dropping the constant factor allows us instead to minimize a sum of *nonlinear least-squares*:

$$X^{MAP} = \underset{X}{\operatorname{argmin}} \sum_i \left\|h_i(X_i) - z_i\right\|^2_{\Sigma_i}$$

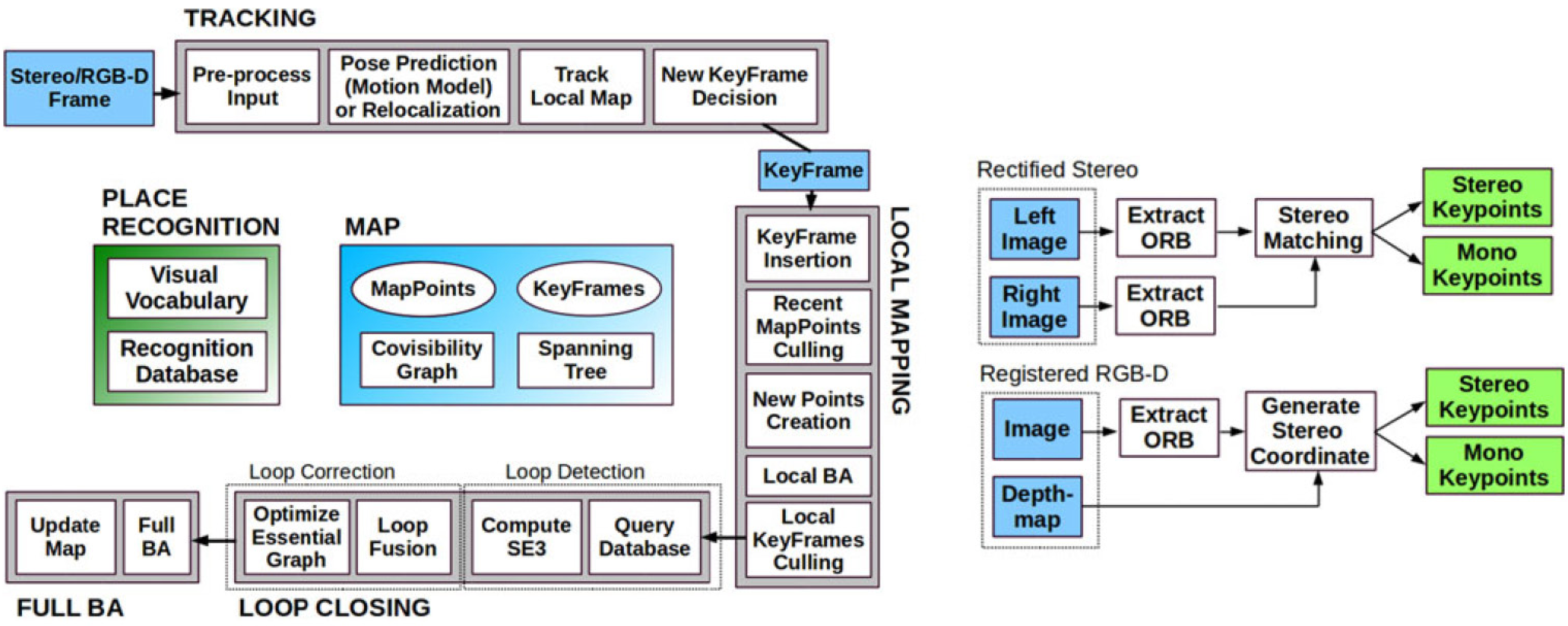
The sparse Jacobian and its factor graph

- The key in modern SLAM is to exploit sparsity
- Factor graphs represent the sparse block structure in the resulting sparse Jacobian $\mathbf{A}$.



$$
\begin{array}{c} \\ \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{array}
\begin{array}{c} \\ [A \,|\, b] = \end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & x_1 & x_2 & x_3 & b \end{array} \\
\left[\begin{array}{ccccc|c}
 & & A_{13} & & & b_1 \\
 & & A_{23} & A_{24} & & b_2 \\
 & & & A_{34} & A_{35} & b_3 \\
A_{41} & & & & & b_4 \\
 & A_{52} & & & & b_5 \\
 & & A_{63} & & & b_6 \\
A_{71} & & A_{73} & & & b_7 \\
A_{81} & & & A_{84} & & b_8 \\
 & A_{92} & & & A_{95} & b_9
\end{array}\right]
\end{array}
$$

ORB-SLAM 2



R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” IEEE Trans. Robot., pp. 1–8, 2017.

Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Image Analysis

Image Segmentation:

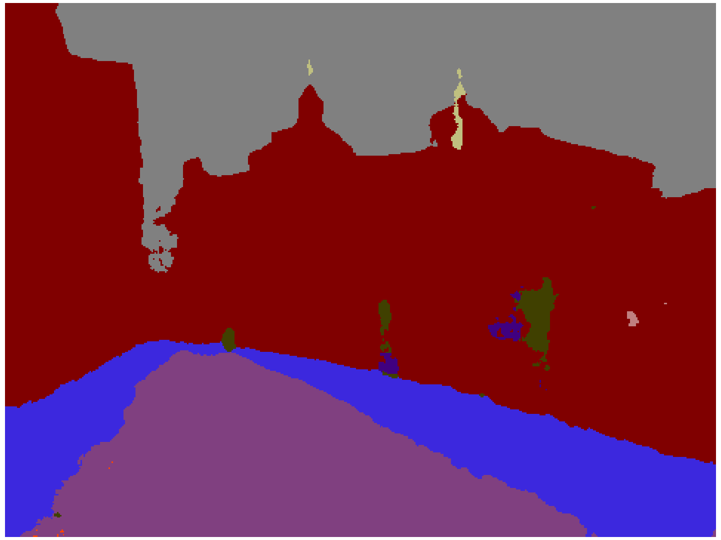
- Thresholding techniques
- Clustering methods for segmentation
- Morphological operations.

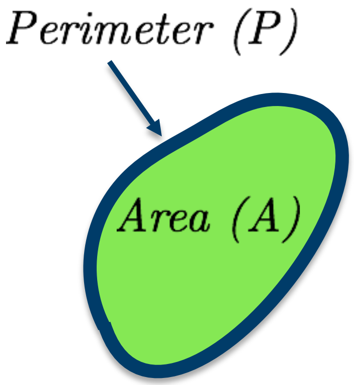
Image feature extraction:

- Feature extraction
- Feature selection.

Introduction to Machine Learning:

- Pattern classification
- Training of classifiers (supervised learning)
- Parametric and non-parametric methods
- Discriminant functions
- Dimensionality reduction.





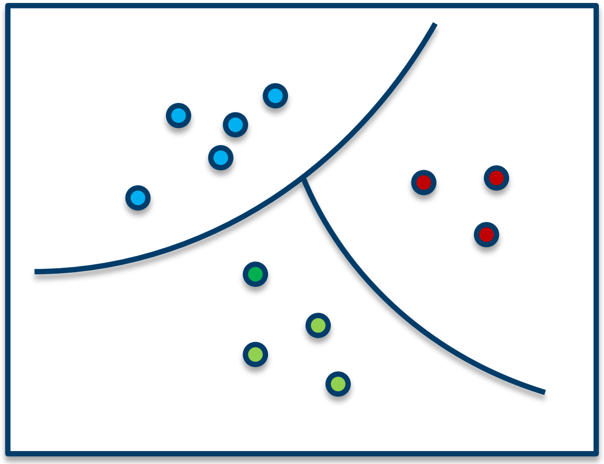
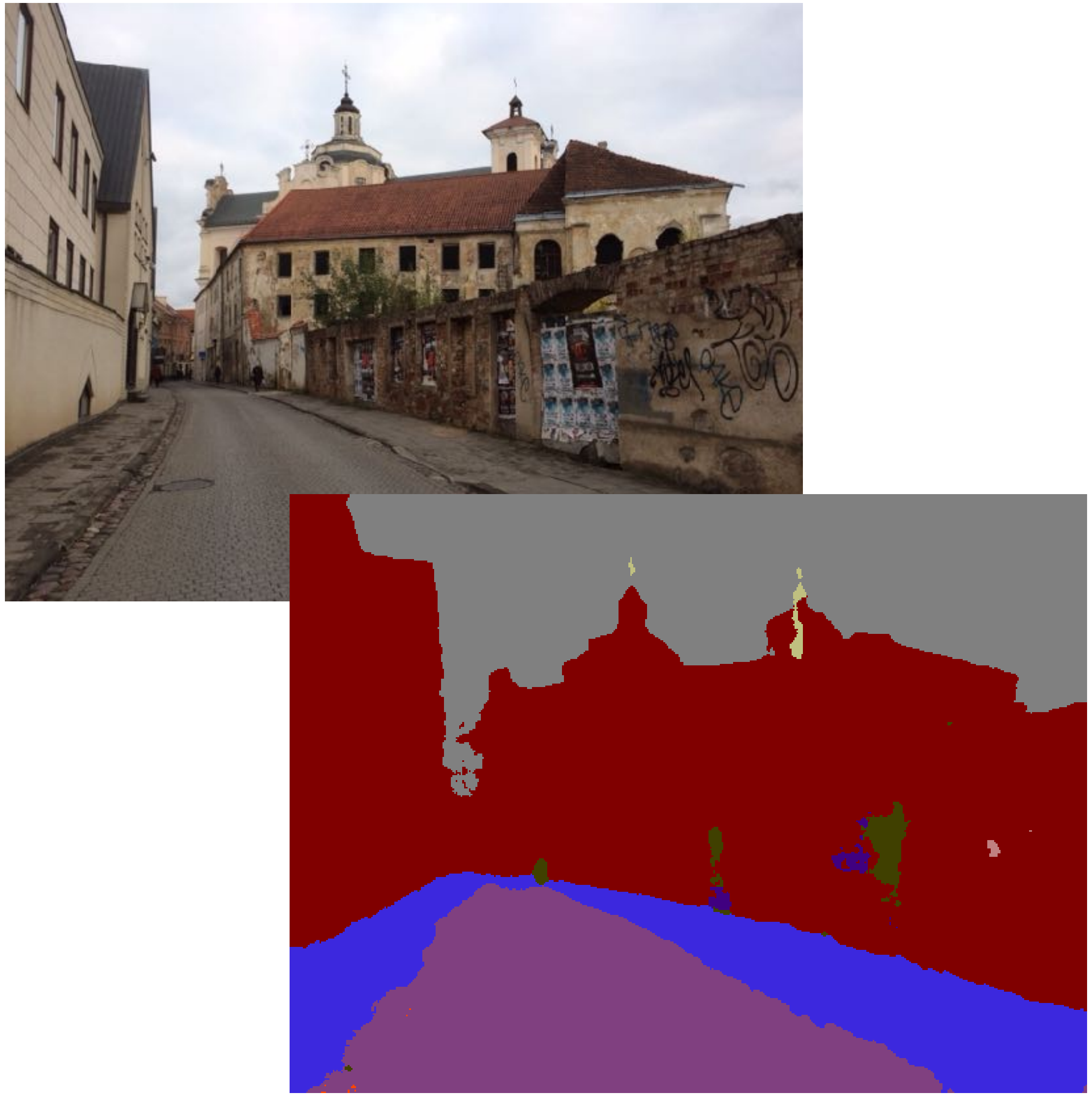


Image Segmentation

Methods:

- Active contours (Snakes, Scissors, Level Sets)
- Split and merge (Watershed, Divisive & agglomerative clustering, Graph-based segmentation)
- Gray level thresholding
- K-means (parametric clustering)
- Mean shift (non-parametric clustering)
- Normalized cuts
- Graph cuts.



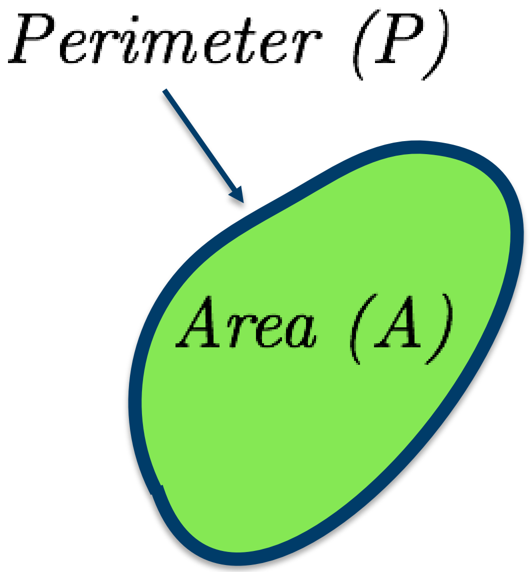
Feature Extraction

The goal is to generate features that exhibit high information-packing properties:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{bmatrix}$$

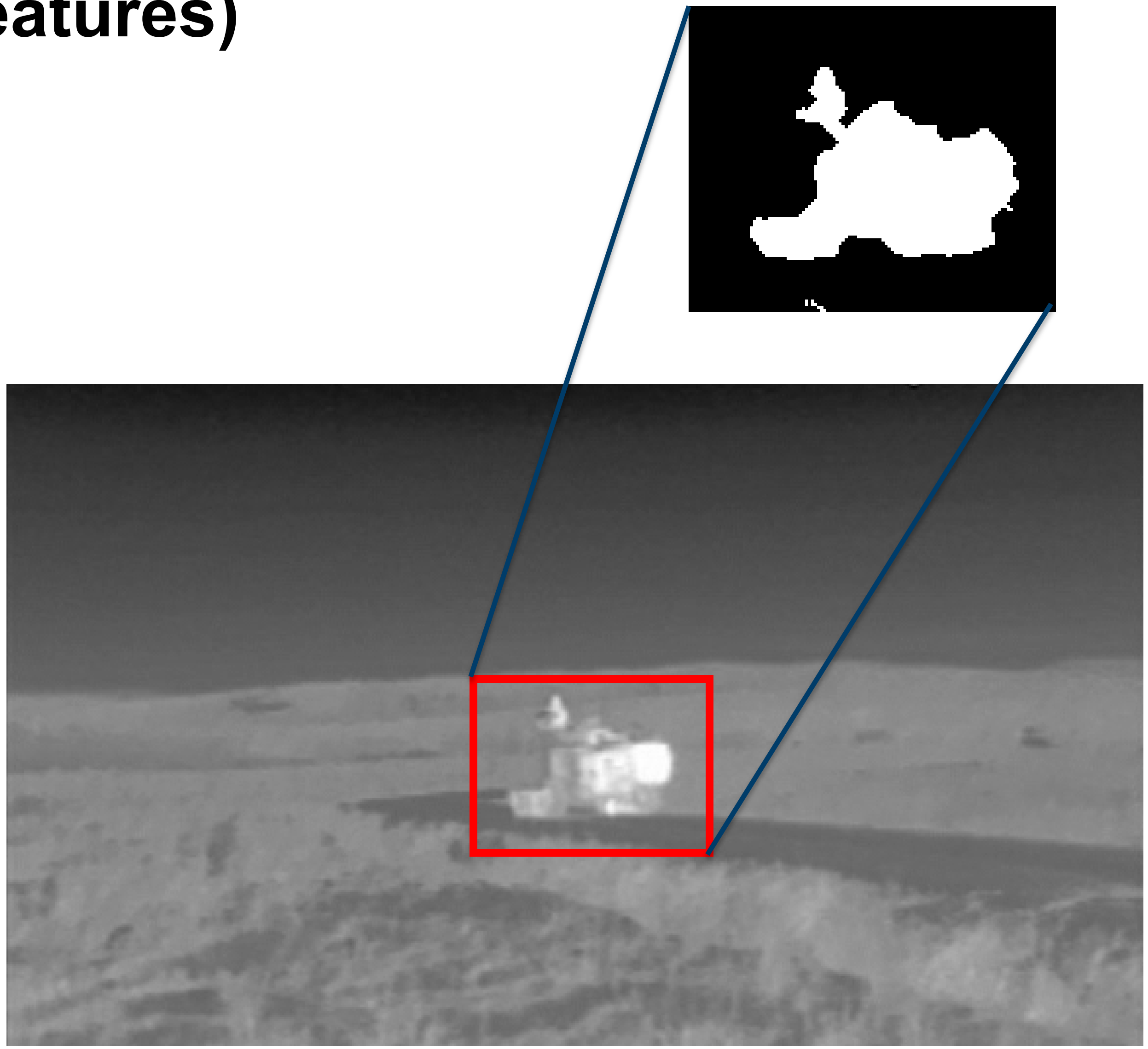
- Extract the information from the raw data that is most relevant for discrimination between the classes
- Extract features with low within-class variability and high between class variability
- Discard redundant information.

- The information in an image f[i,j] must be reduced to enable reliable classification (generalization)
- A 64x64 image → 4096-dimensional feature space!



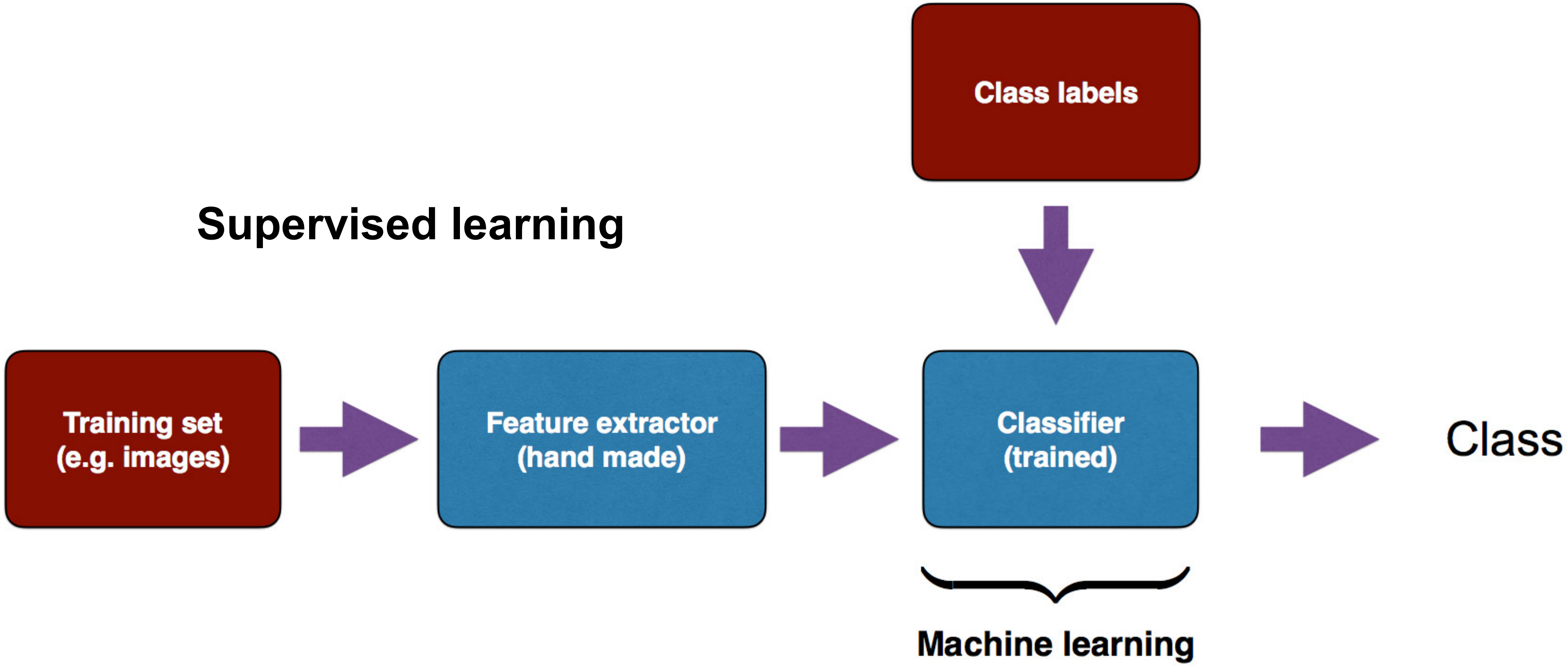
Feature types (regional features)

- Colour features
- Shape features
- Histogram (texture) features:
  - Mean gray level
  - Variance
  - Skewness
  - Kurtosis
  - Entropy
  - …



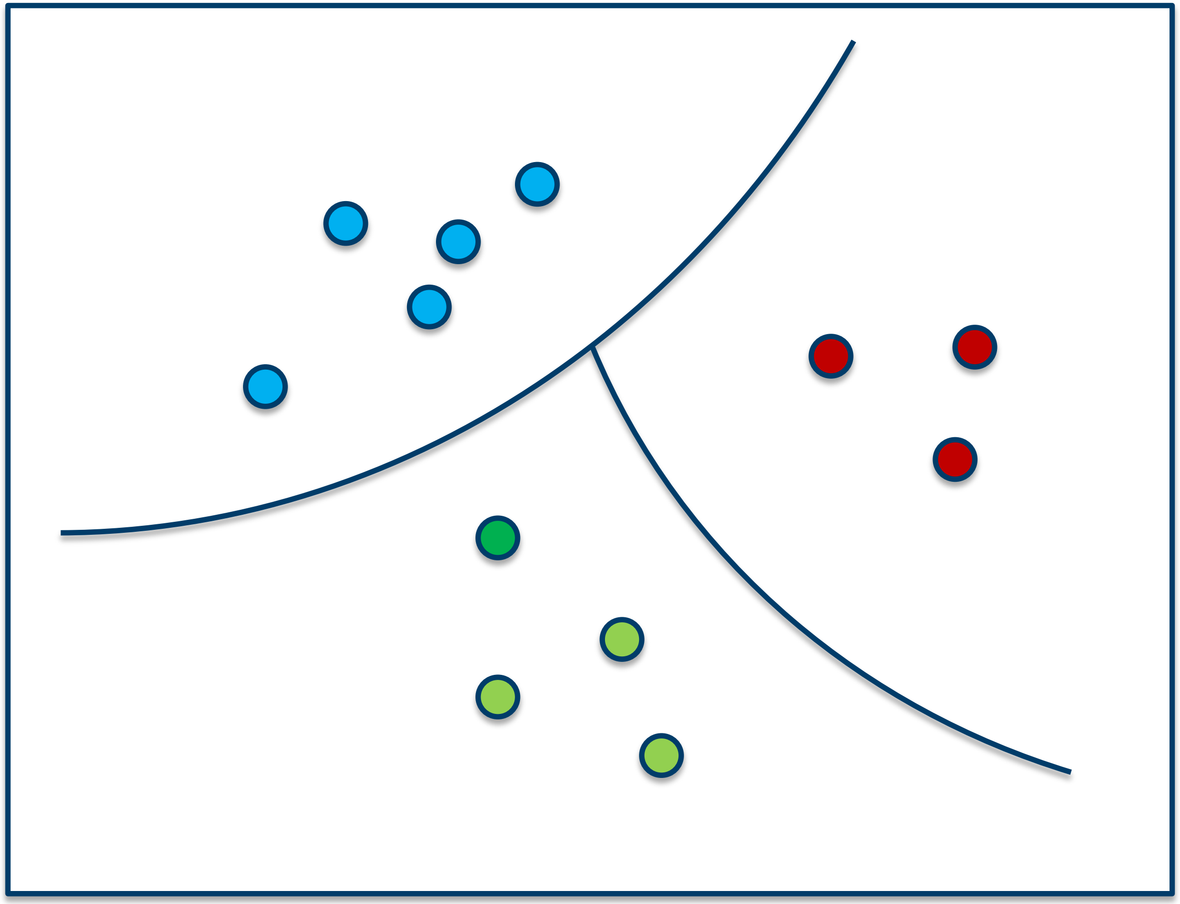
Introduction to Machine learning

Discrimination between classes (pattern recognition, classification)



Classifiers and training methods

- Bayes classifier
- Nearest-neighbors and K-nearest-neighbors
- Parzen windows
- Linear and higher order discriminant functions
- Neural nets
- Support Vector Machines (SVM)
- Decision trees
- Random forest
- …



Lectures 2019

IMAGE FORMATION, PROCESSING AND FEATURES

- **Image formation**
  - Light, cameras, optics and color
  - The perspective camera model
  - Basic projective geometry
- **Image processing**
  - Image filtering
  - Image pyramids
  - Laplace blending
- **Feature detection**
  - Line features
  - Local keypoint features
  - Robust estimation with RANSAC
- **Feature matching**
  - From keypoints to feature correspondences
  - Feature descriptors
  - Feature matching
  - Estimating homographies from feature correspondences

WORLD GEOMETRY AND 3D

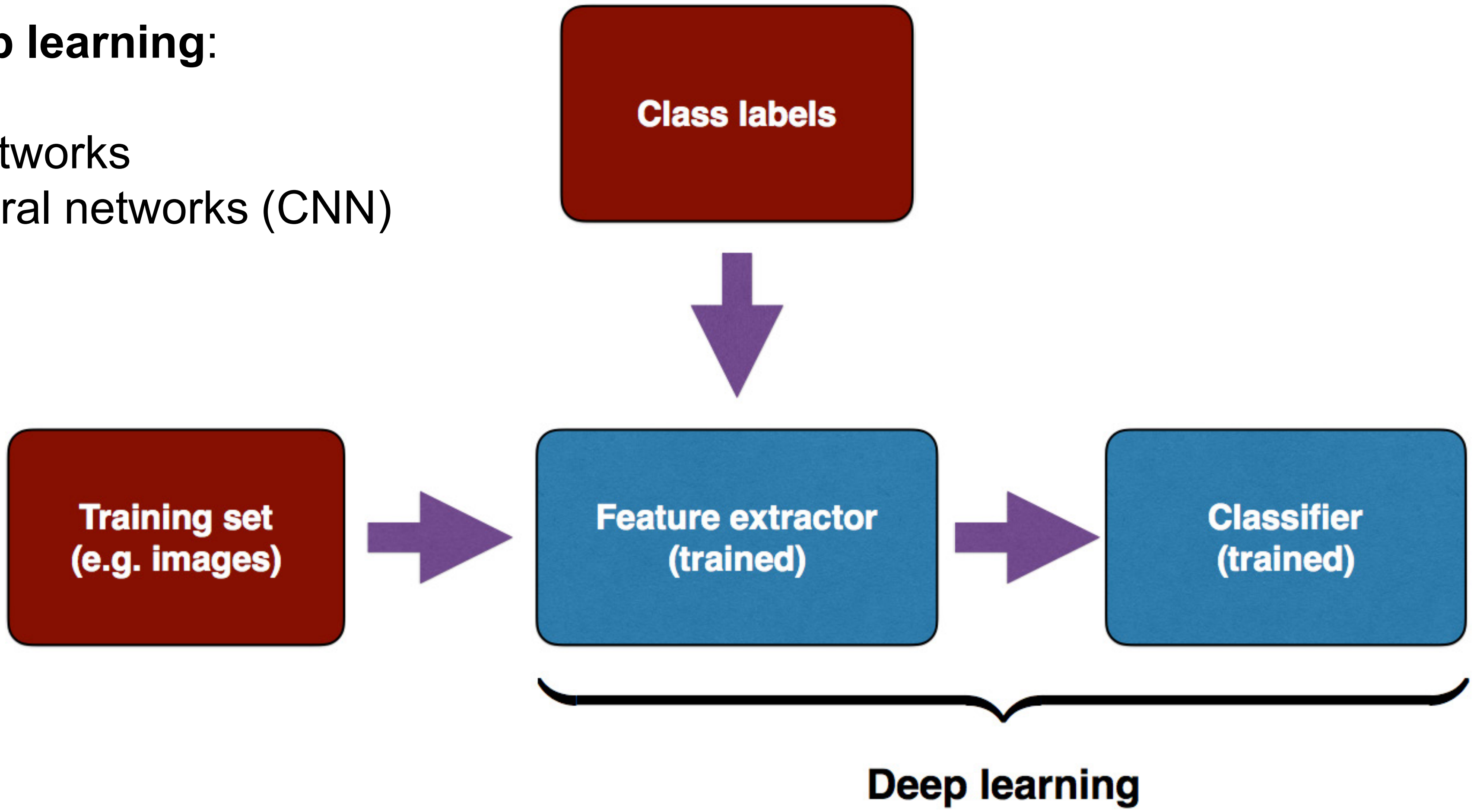
- **3D pose representation**
  - Orientation in 3D
  - Pose in 3D
  - The perspective camera model revisited
- **Single-View geometry**
  - Pose from a known 3D map
  - An introduction to nonlinear least squares
  - Optimization over poses
  - Nonlinear pose estimation
- **Stereo imaging**
  - Basic epipolar geometry
  - Stereo imaging
  - Stereo processing
- **Two-view geometry**
  - Epipolar geometry
  - Triangulation
  - Triangulation by minimizing reprojection error
  - Pose from epipolar geometry
- **Multiple-view geometry**
  - Multiple-view geometry
  - Structure from motion
  - Multiple-view stereo
- **Visual SLAM**
  - Introduction to Visual SLAM
  - Map optimization
  - ORB-SLAM

SCENE ANALYSIS

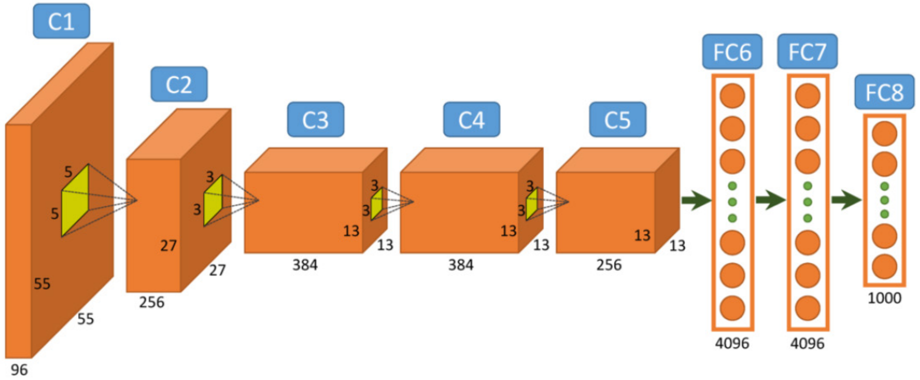
- **Image analysis**
  - Image segmentation
  - Image feature extraction
  - Introduction to machine learning
- **Object recognition**
  - Deep learning

Detection and recognition with deep learning

Introduction to deep learning:
- Deep learning
- Artificial neural networks
- Convolutional neural networks (CNN)



Deep Learning for Object Recognition



«Ship»

Millions of images

Millions of parameters

Thousands of classes