Lecture 7.3
Pose from epipolar geometry

Thomas Opsahl

Introduction

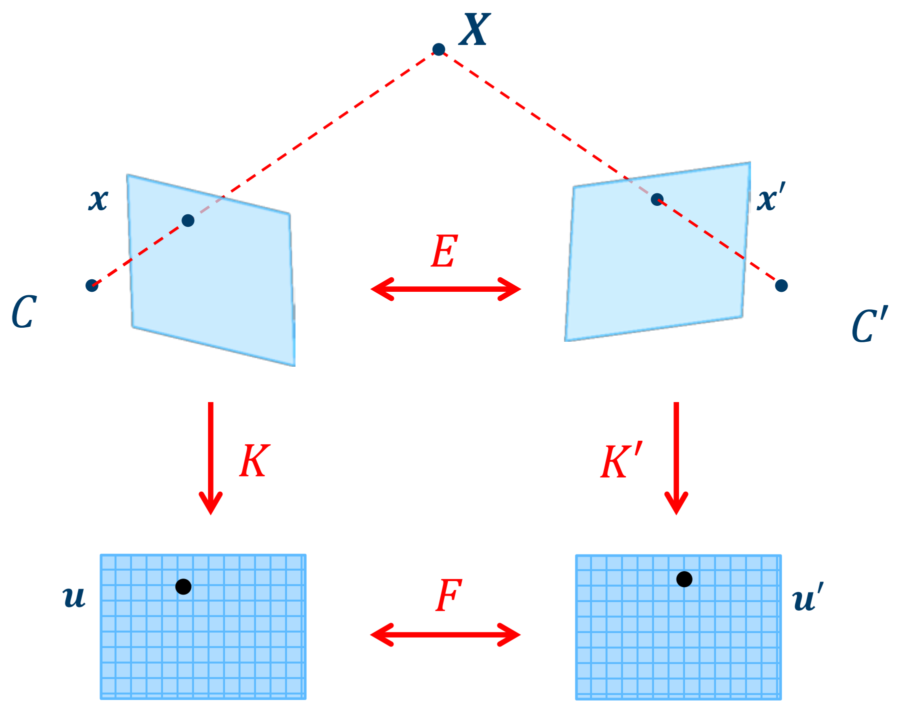
- Representing epipolar geometry
  - The essential matrix $E = [\boldsymbol{t}]_\times R$

    $$\tilde{\boldsymbol{x}}'^T E \tilde{\boldsymbol{x}} = 0$$

  - The fundamental matrix $F = K'^{-T} E K^{-1}$

    $$\tilde{\boldsymbol{u}}'^T F \tilde{\boldsymbol{u}} = 0$$

- Estimating epipolar geometry
  - F from 7 or 8 2D correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}_i{}'$
  - E from 5 2D correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i{}'$



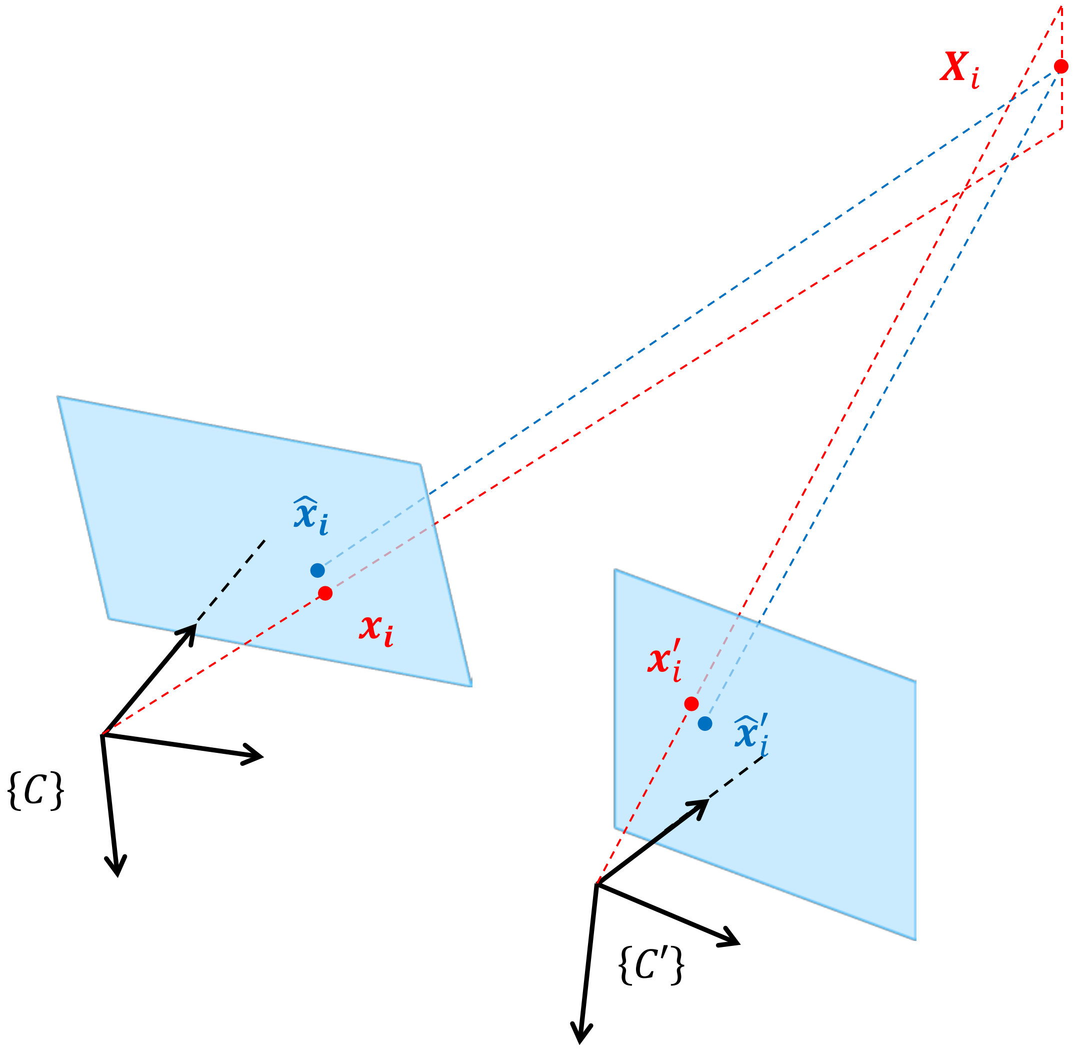
Introduction

- Representing epipolar geometry
  - The essential matrix $E = [\boldsymbol{t}]_\times R$
    $$\tilde{\boldsymbol{x}}'^T E \tilde{\boldsymbol{x}} = 0$$
  - The fundamental matrix $F = K'^{-T} E K^{-1}$
    $$\tilde{\boldsymbol{u}}'^T F \tilde{\boldsymbol{u}} = 0$$

- Estimating epipolar geometry
  - F from 7 or 8 2D correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}_i{}'$
  - E from 5 2D correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i{}'$

- Exploiting epipolar geometry
  - 3D reconstruction of the scene by triangulation when camera matrices are known

- Now we will look at another way to make use of the epipolar geometry



Recall the transformations of projective space

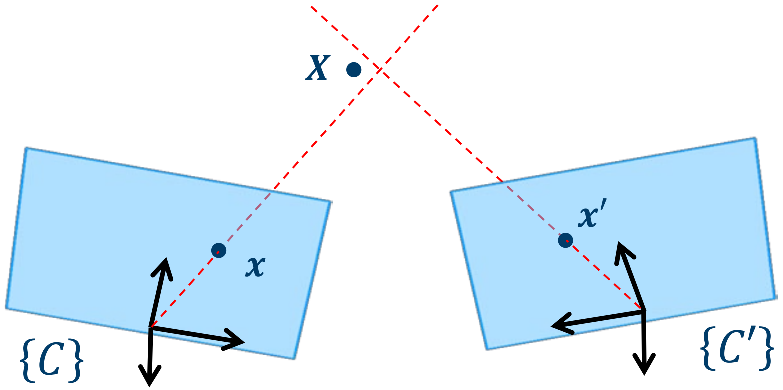
Transformation of $\mathbb{P}^3$	Matrix	#DoF	Preserves
Translation	$\begin{bmatrix} I & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	3	Orientation + all below
Euclidean	$\begin{bmatrix} R & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	6	Volumes, volume ratios, lengths + all below
Similarity	$\begin{bmatrix} sR & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$	7	Angles + all below
Affine	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$	12	Parallelism of planes, The plane at infinity + all below
Homography /projective	$\begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix}$	15	Intersection and tangency of surfaces in contact, straight lines

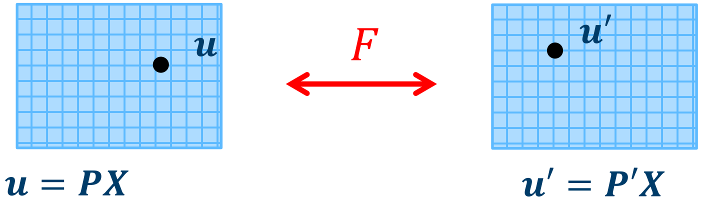
Pose from epipolar geometry

- One of the most important results in computer vision is that it is possible to determine the camera matrices P and P' that correspond to a given fundamental matrix F
  - Not uniquely, but up to a projective ambiguity, so lengths, angles or parallel lines/surfaces are not preserved

- If H is a projective transformation of 3-space, then the fundamental matrix corresponding to the pair of camera matrices (P, P') is the same as to that of (HP, HP')

- This can still be used to estimate the structure of the scene by triangulation, but the reconstructed scene would also suffer from a projective ambiguity





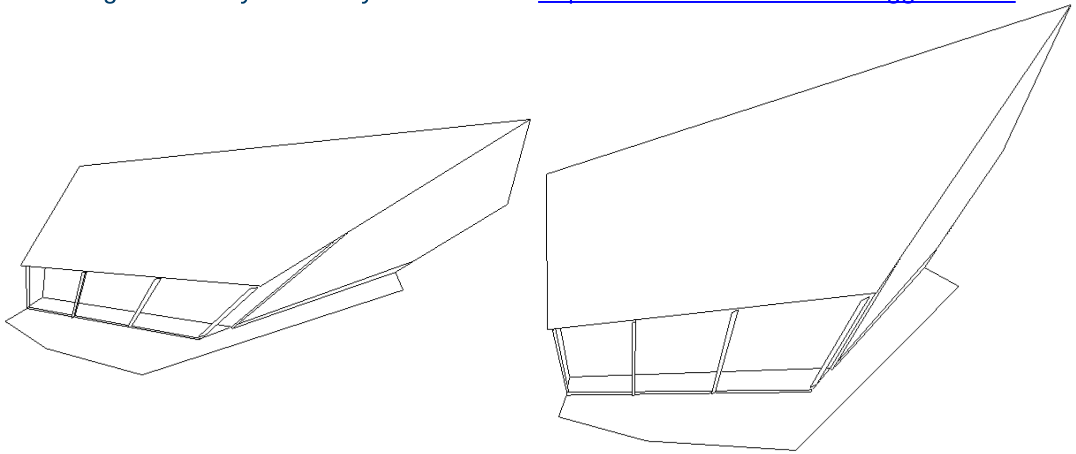
Pose from epipolar geometry

- One of the most important results in computer vision is that it is possible to determine the camera matrices P and P' that correspond to a given fundamental matrix F
  - Not uniquely, but up to a projective ambiguity, so lengths, angles or parallel lines/surfaces are not preserved

- If H is a projective transformation of 3-space, then the fundamental matrix corresponding to the pair of camera matrices (P, P') is the same as to that of (HP, HP')

- This can still be used to estimate the structure of the scene by triangulation, but the reconstructed scene would also suffer from a projective ambiguity

Images courtesy of Hartley & Zisserman http://www.robots.ox.ac.uk/~vgg/hzbook/



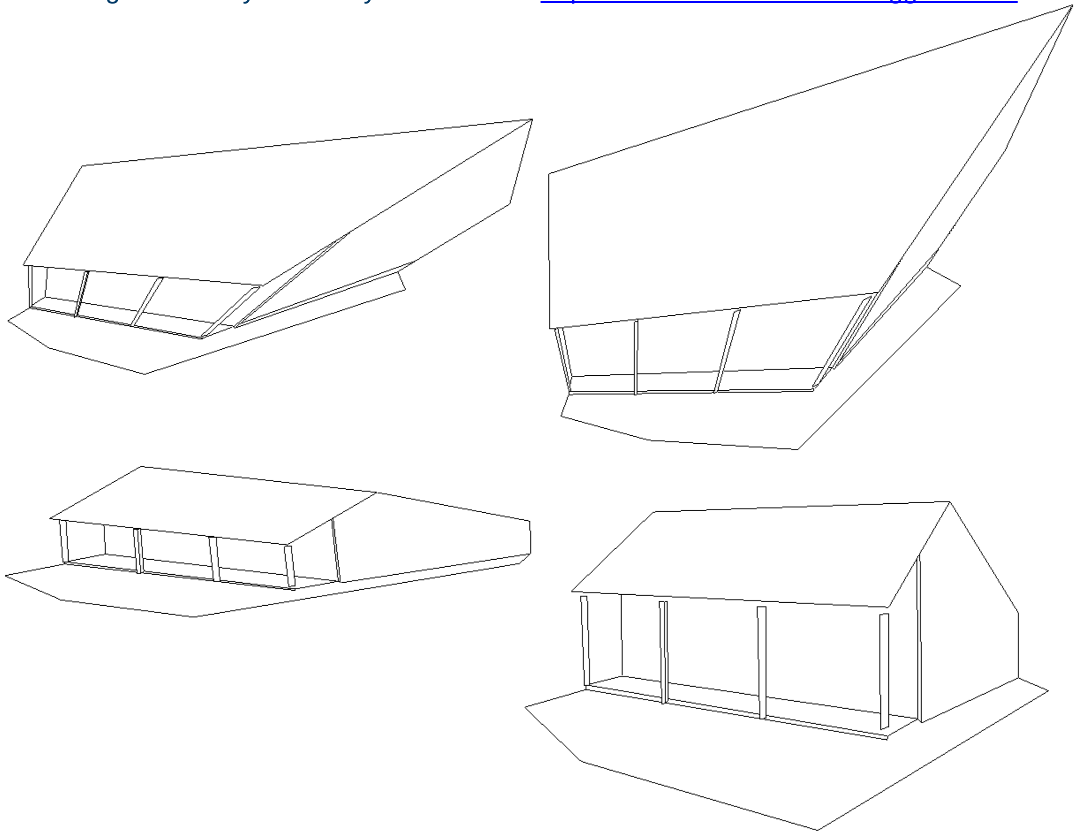
Pose from epipolar geometry

- One of the most important results in computer vision is that it is possible to determine the camera matrices P and P' that correspond to a given fundamental matrix F
  - Not uniquely, but up to a projective ambiguity, so lengths, angles or parallel lines/surfaces are not preserved

- If H is a projective transformation of 3-space, then the fundamental matrix corresponding to the pair of camera matrices (P, P') is the same as to that of (HP, HP')

- This can still be used to estimate the structure of the scene by triangulation, but the reconstructed scene would also suffer from a projective ambiguity
  - By adding knowledge about the scene the ambiguity can be restricted to affine or even metric

Images courtesy of Hartley & Zisserman http://www.robots.ox.ac.uk/~vgg/hzbook/



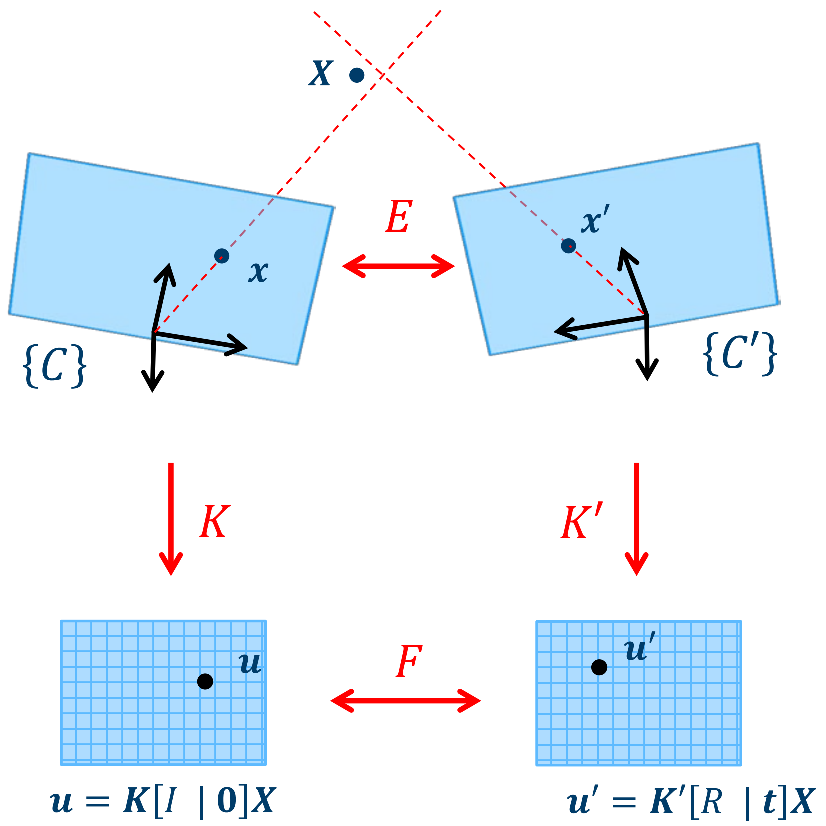
Pose from epipolar geometry

- If we restrict ourselves to calibrated cameras, the ambiguity gets restricted as well

- In the calibrated case, we can estimate and represent the epipolar geometry in terms of the essential matrix E

- By estimation, E will be a homogeneous matrix only restricted by
$$\tilde{\boldsymbol{x}}'^T E \tilde{\boldsymbol{x}} = 0$$

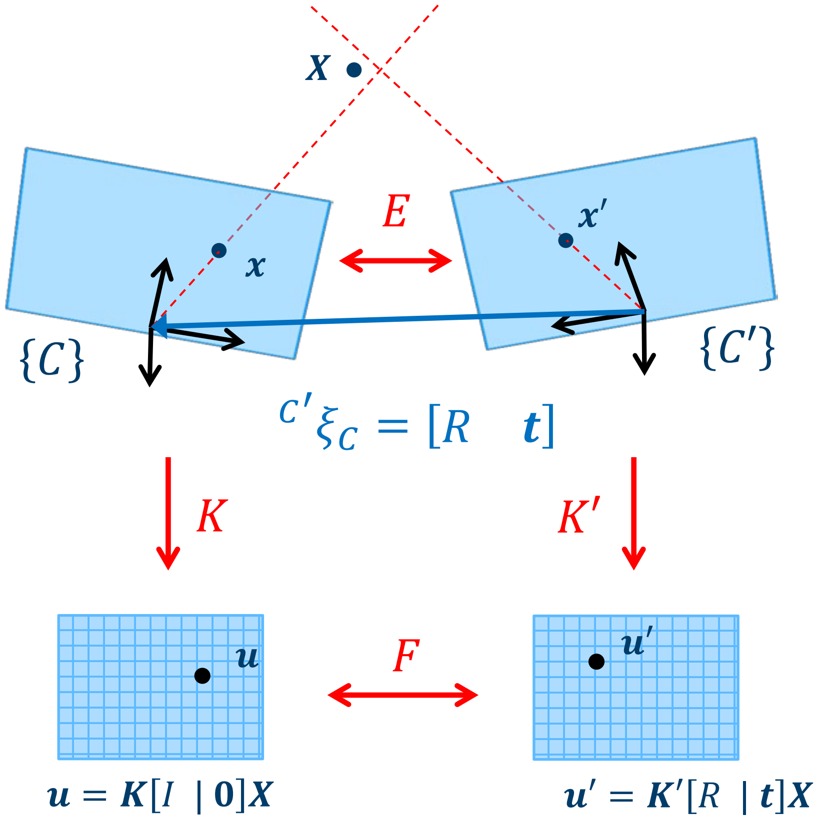
- But we also know that we can construct E non-homogeneously as $E = [\boldsymbol{t}]_\times R$



Pose from epipolar geometry

- In 1981 H. C Longuet-Higgins* proved that one could recover the relative pose ${}^{C'}\xi_C = [R \quad \boldsymbol{t}]$ from the essential matrix up to the scale of $\boldsymbol{t}$

- He argued that, up to the scale of $\boldsymbol{t}$, there are 4 theoretical solutions, but only 1 for which the scene points would be in front of both cameras
  - This additional constraint has later been named the *cheirality constraint*



* H. C Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, 1981

Pose from epipolar geometry

- Since we only can estimate E up to scale, we can always rescale it so that the SVD of E has the form

$$E = UDV^T = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \boldsymbol{u}_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \boldsymbol{v}_3^T \end{bmatrix}$$

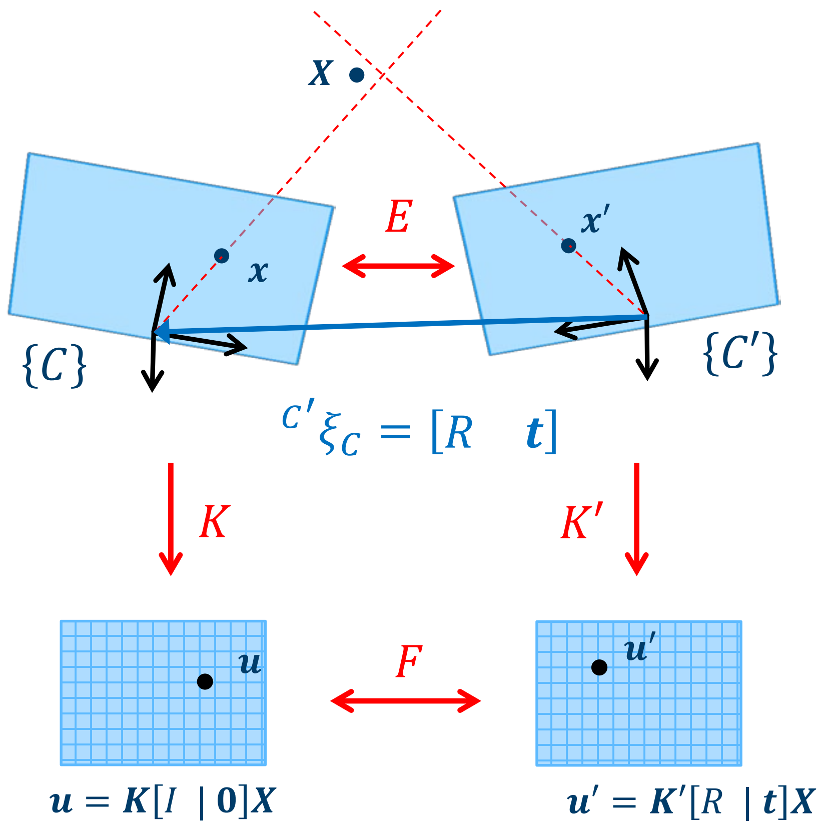
  where $det(U) = det(V) = 1$

- Then one can show that

$$R \in \{UWV^T, UW^TV^T\}$$
$$\boldsymbol{t} = \pm\lambda \boldsymbol{u}_3; \;\; \lambda \in \mathbb{R}\backslash 0$$

  where

$$W = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Pose from epipolar geometry

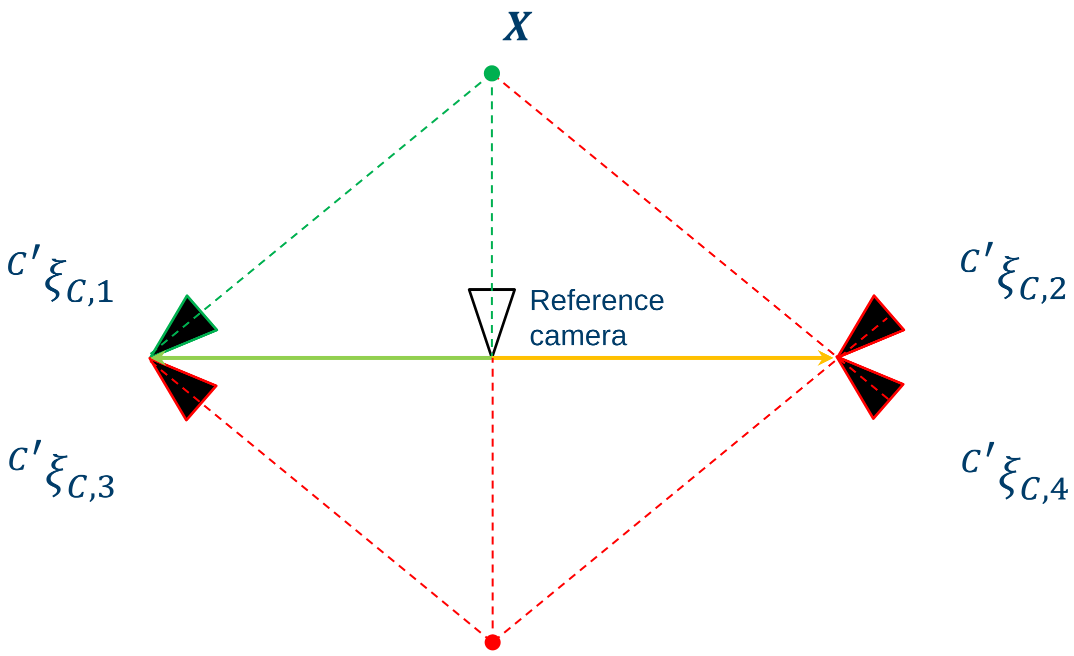
- So the 4 candidate poses are

$$
\begin{aligned}
{}^{C'}\xi_{C,1} &= [UWV^T \quad \boldsymbol{u}_3] \\
{}^{C'}\xi_{C,2} &= [UWV^T \quad -\boldsymbol{u}_3] \\
{}^{C'}\xi_{C,3} &= [UW^TV^T \quad \boldsymbol{u}_3] \\
{}^{C'}\xi_{C,4} &= [UW^TV^T \quad -\boldsymbol{u}_3]
\end{aligned}
$$

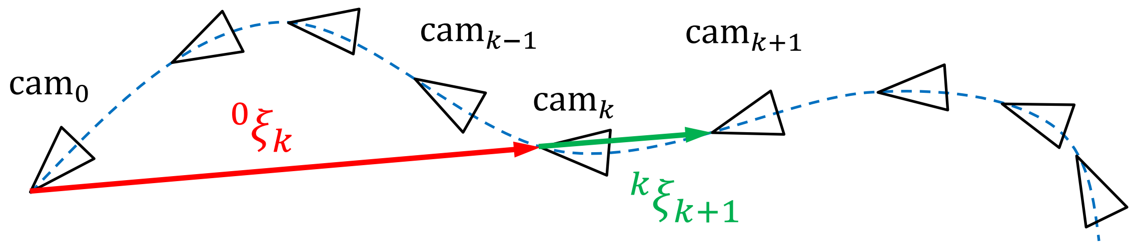
- Their relation is shown in the figure for the case when ${}^{C'}\xi_{C,1}$ is the correct pose

- In general there is no way of knowing the correct candidate without imposing the cheirality constraint

- In theory it suffices to triangulate a single scene point $\boldsymbol{X}$ in order to determine the correct pose, but for robustness several points could be checked



Visual odometry



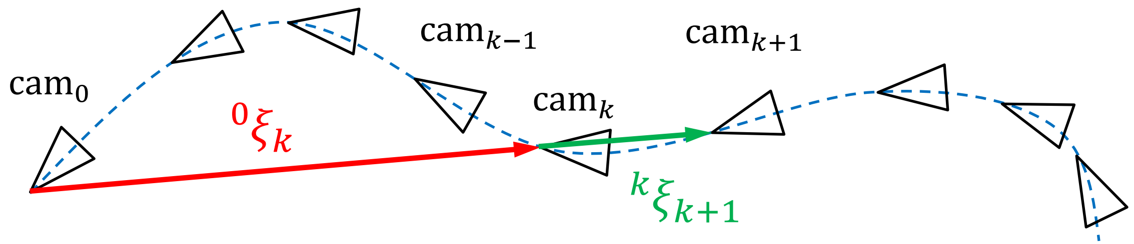
- Based on what we now know it is possible to do visual odometry, i.e. estimating the motion of a single camera from captured images

- The algorithm could look something like this

Visual odometry from 2D-correspondences

1. Capture new frame img_{k+1}
2. Extract and match features between img_{k+1} and img_k
3. Estimate the essential matrix $E_{k,k+1}$
4. Decompose the $E_{k,k+1}$ into ${}^kR_{k+1}$ and ${}^k\boldsymbol{t}_{k+1}$ to get the relative pose
$$ {}^k\xi_{k+1} = [{}^kR_{k+1} \quad {}^k\boldsymbol{t}_{k+1}] $$
5. Calculate the pose of camera $k+1$ relative to the first camera
$$ {}^0\xi_{k+1} = {}^0\xi_k \, {}^k\xi_{k+1} $$

Visual odometry



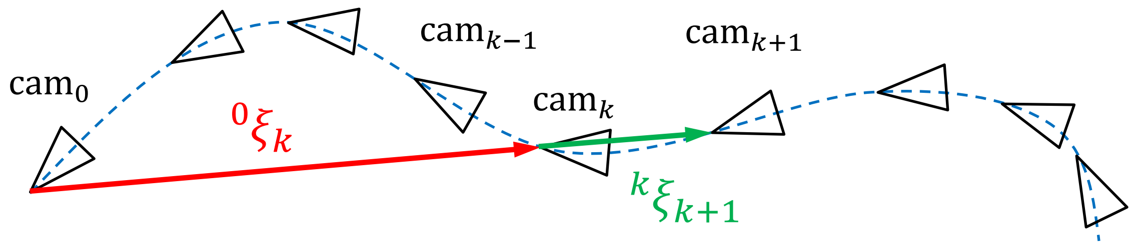
- Based on what we now know it is possible to do visual odometry, i.e. estimating the motion of a single camera from captured images

- The algorithm could look something like this
  - Neglects the unknown scale of ${}^k\boldsymbol{t}_{k-1}$
  - We should set $\|{}^1\boldsymbol{t}_0\| = 1$ and scale the other translations accordingly

Visual odometry from 2D-correspondences

1. Capture new frame img_{k+1}
2. Extract and match features between img_{k+1} and img_k
3. Estimate the essential matrix $E_{k,k+1}$
4. Decompose the $E_{k,k+1}$ into ${}^kR_{k+1}$ and ${}^k\boldsymbol{t}_{k+1}$ to get the relative pose
$$ {}^k\xi_{k+1} = [{}^kR_{k+1} \quad {}^k\boldsymbol{t}_{k+1}] $$
5. Calculate the pose of camera $k+1$ relative to the first camera
$$ {}^0\xi_{k+1} = {}^0\xi_k \, {}^k\xi_{k+1} $$

Visual odometry



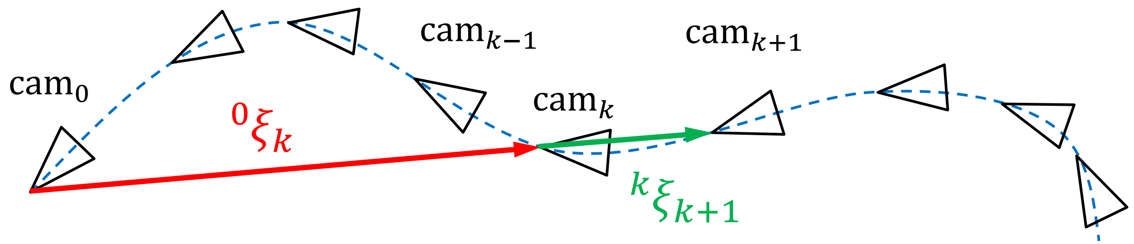
- Based on what we now know it is possible to do visual odometry, i.e. estimating the motion of a single camera from captured images

- A better visual odometry algorithm can look like this

Visual odometry from 2D-correspondences

1. Capture new frame img_{k+1}
2. Extract and match features between img_{k+1} and img_k
3. Estimate the essential matrix $E_{k,k+1}$
4. Decompose the $E_{k,k+1}$ into ${}^kR_{k+1}$ and ${}^k\boldsymbol{t}_{k+1}$ to get the relative pose
$${}^k\xi_{k+1} = [{}^kR_{k+1} \quad {}^k\boldsymbol{t}_{k+1}]$$
5. Compute $\|{}^k\boldsymbol{t}_{k+1}\|$ from $\|{}^{k-1}\boldsymbol{t}_k\|$ and rescale ${}^k\boldsymbol{t}_{k+1}$ accordingly
6. Calculate the pose of camera $k+1$ relative to the first camera
$${}^0\xi_{k+1} = {}^0\xi_k \, {}^k\xi_{k+1}$$

Visual odometry

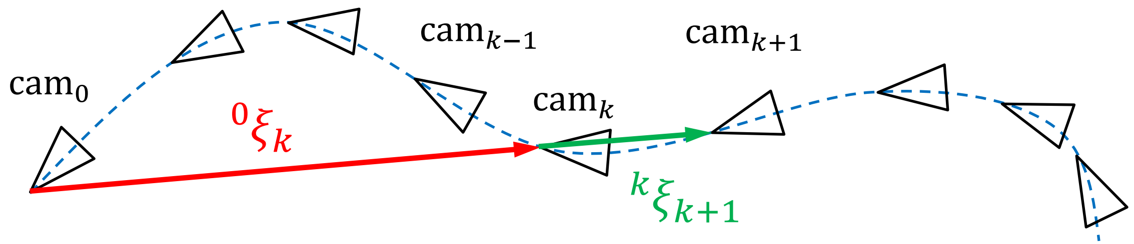


- Based on what we now know it is possible to do visual odometry, i.e. estimating the motion of a single camera from captured images
- A better visual odometry algorithm can look like this
  - How to compute $\|{}^{k+1}\boldsymbol{t}_k\|$ from $\|{}^k\boldsymbol{t}_{k-1}\|$?

Visual odometry from 2D-correspondences

1. Capture new frame img_{k+1}
2. Extract and match features between img_{k+1} and img_k
3. Estimate the essential matrix $E_{k,k+1}$
4. Decompose the $E_{k,k+1}$ into ${}^kR_{k+1}$ and ${}^k\boldsymbol{t}_{k+1}$ to get the relative pose
$${}^k\xi_{k+1} = [{}^kR_{k+1} \quad {}^k\boldsymbol{t}_{k+1}]$$
5. Compute $\|{}^k\boldsymbol{t}_{k+1}\|$ from $\|{}^{k-1}\boldsymbol{t}_k\|$ and rescale ${}^k\boldsymbol{t}_{k+1}$ accordingly
6. Calculate the pose of camera $k+1$ relative to the first camera
$${}^0\xi_{k+1} = {}^0\xi_k\,{}^k\xi_{k+1}$$

Visual odometry



- Based on what we now know it is possible to do visual odometry, i.e. estimating the motion of a single camera from captured images

- A better visual odometry algorithm can look like this
  - How to compute $\|{}^{k+1}\boldsymbol{t}_k\|$ from $\|{}^k\boldsymbol{t}_{k-1}\|$?
  - Determine two scene points ${}^k\boldsymbol{X}_{k-1,k}$ and ${}^k\boldsymbol{X}'_{k-1,k}$ by triangulation of two 2D-correspondences ${}^{k-1}\boldsymbol{x} \leftrightarrow {}^k\boldsymbol{x}$ and ${}^{k-1}\boldsymbol{x}' \leftrightarrow {}^k\boldsymbol{x}'$
  - Determine the same two scene points ${}^k\boldsymbol{X}_{k,k+1}$ and ${}^k\boldsymbol{X}'_{k,k+1}$ by triangulation of two 2D-correspondences ${}^k\boldsymbol{x} \leftrightarrow {}^{k+1}\boldsymbol{x}$ and ${}^k\boldsymbol{x}' \leftrightarrow {}^{k+1}\boldsymbol{x}'$
  - Then

$$\frac{\left\|{}^{k-1}\boldsymbol{t}_k\right\|}{\left\|{}^k\boldsymbol{t}_{k+1}\right\|} = \frac{\left\|{}^k\boldsymbol{X}_{k-1,k} - {}^k\boldsymbol{X}'_{k-1,k}\right\|}{\left\|{}^k\boldsymbol{X}_{k,k+1} - {}^k\boldsymbol{X}'_{k,k+1}\right\|}$$

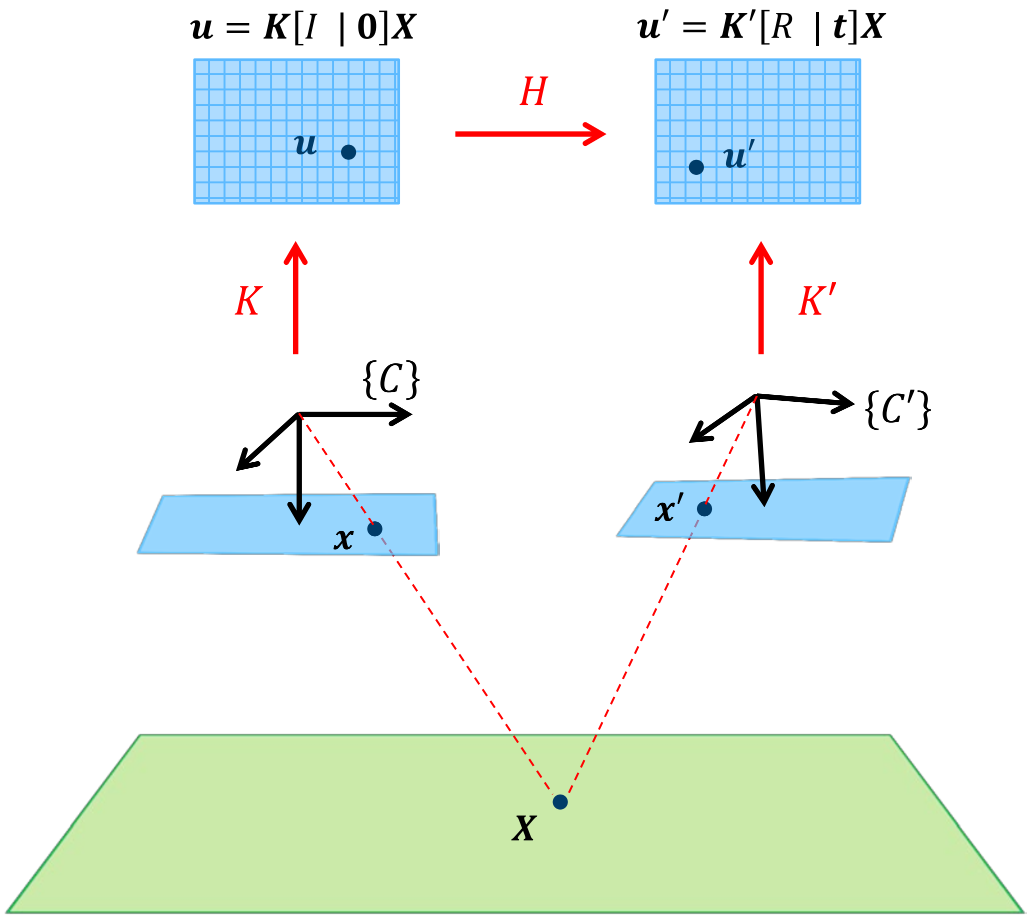
Visual odometry from 2D-correspondences

1. Capture new frame img_{k+1}
2. Extract and match features between img_{k+1} and img_k
3. Estimate the essential matrix $E_{k,k+1}$
4. Decompose the $E_{k,k+1}$ into ${}^kR_{k+1}$ and ${}^k\boldsymbol{t}_{k+1}$ to get the relative pose
$${}^k\xi_{k+1} = [{}^kR_{k+1} \quad {}^k\boldsymbol{t}_{k+1}]$$
5. Compute $\|{}^k\boldsymbol{t}_{k+1}\|$ from $\|{}^{k-1}\boldsymbol{t}_k\|$ and rescale ${}^k\boldsymbol{t}_{k+1}$ accordingly
6. Calculate the pose of camera $k+1$ relative to the first camera
$${}^0\xi_{k+1} = {}^0\xi_k\,{}^k\xi_{k+1}$$

Planar scene

- When the scene is planar, it is not possible to estimate the epipolar geometry between two views from 2D-correspondences
  - In the case of an almost planar scene, the estimation is likely to be ill-conditioned

- We know that for this case the relationship between image points and scene points can be described by homographies



Planar scene

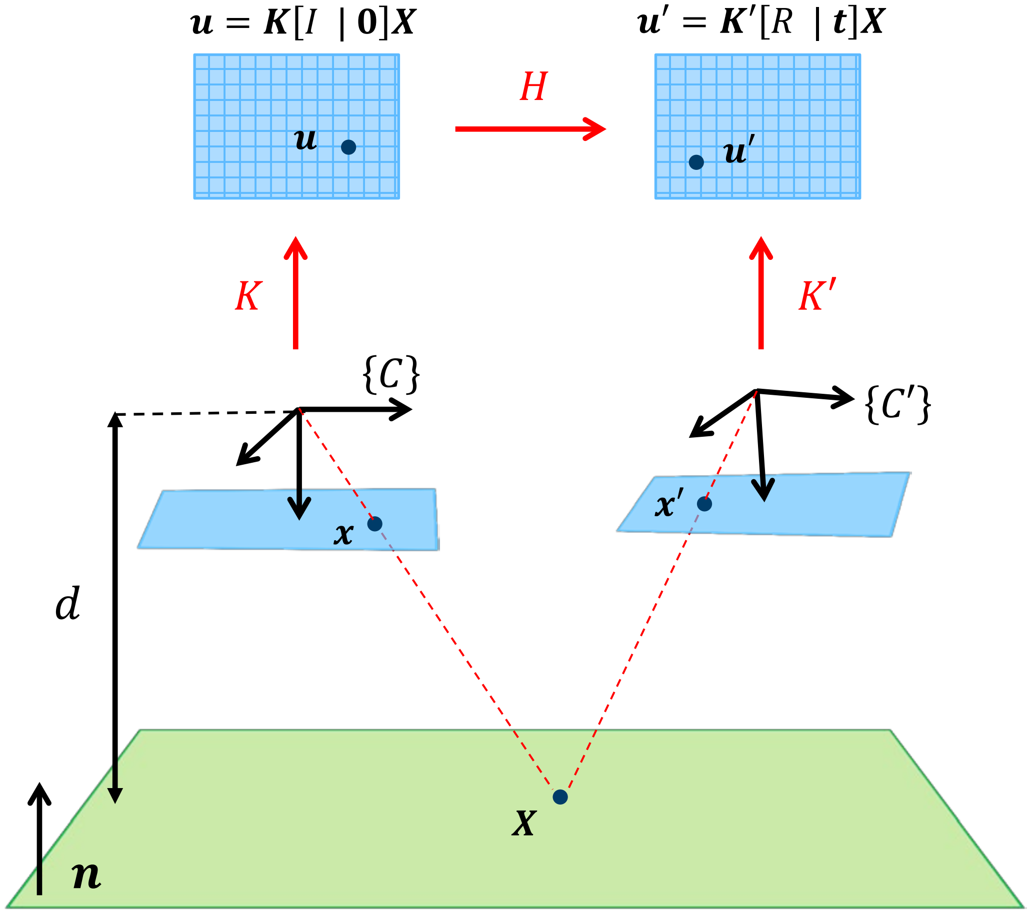
- When the scene is planar, it is not possible to estimate the epipolar geometry between two views from 2D-correspondences
  - In the case of an almost planar scene, the estimation is likely to be ill-conditioned

- We know that for this case the relationship between image points and scene points can be described by homographies

- If the relative pose between views are ${}^{C'}\xi_C = [R \quad \boldsymbol{t}]$, then one can show that the homography between views must be given by

$$H = K'(R - \boldsymbol{t}\boldsymbol{n}^T/d)K^{-1}$$

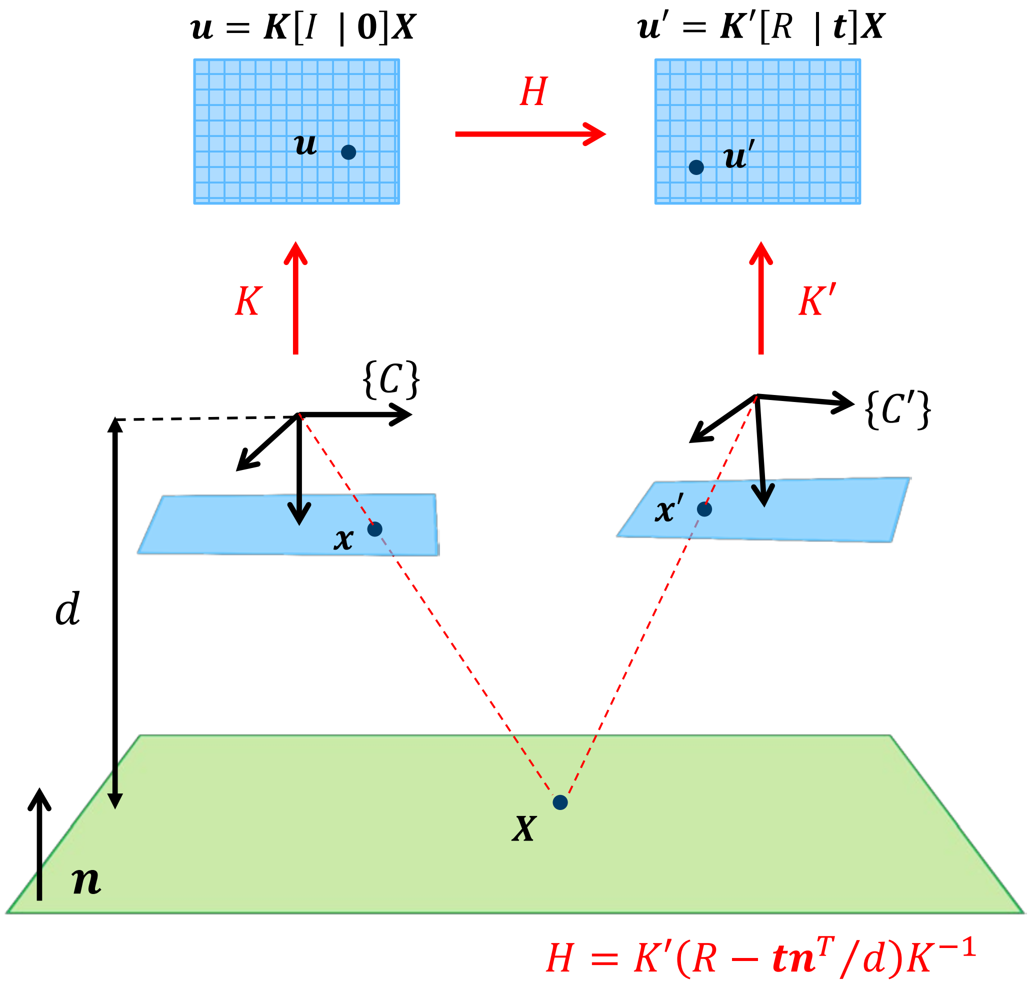
  where $\boldsymbol{n}$ is the normal vector of the plane and d is the depth of the plane relative to $\{C\}$



Planar scene

- Based on the expression $H = K'(R - \boldsymbol{t}\boldsymbol{n}^T/d)K^{-1}$ it is possible to estimate $(R, \boldsymbol{n}, \boldsymbol{t}/d)$ from a known homography in a process known as homography decompositon
  – So for situations where we know the plane depth d, we get the relative pose ${}^{C'}\xi_C = [R \quad \boldsymbol{t}]$

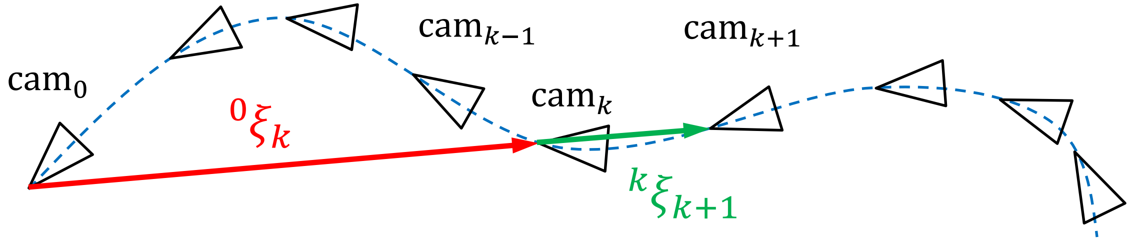
- In the 2007 report *Deeper understanding of the homography decomposition for vision-based control*, Ezio Malis & Manuel Vargas derive that the decomposition problem has 4 analytical solutions
  – Two solutions can be invalidated by requiring points to be in front of the cameras
  – With some knowledge about the normal vector $\boldsymbol{n}$ it is often possible to eliminate one out of the two remaining solutions
  – OpenCV – cv::decomposeHomographyMat

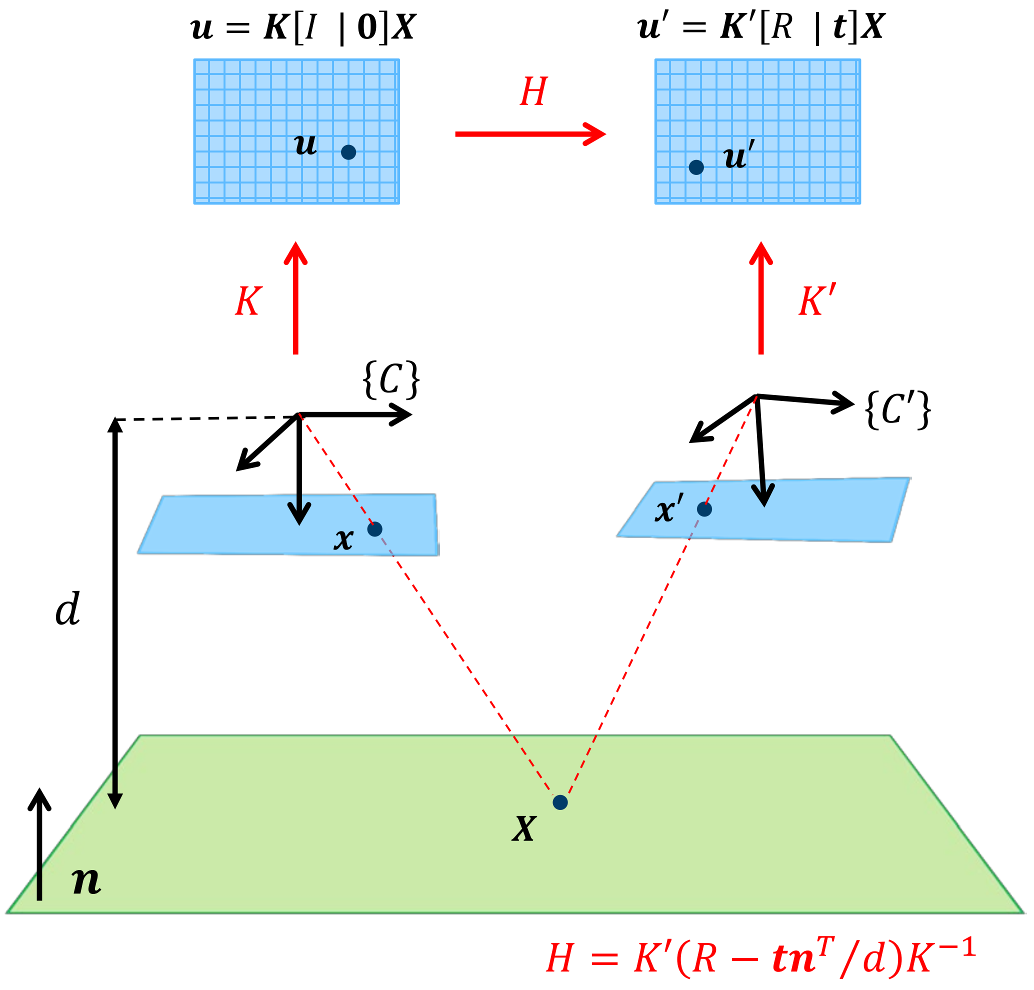


Planar scene

Visual odometry from 2D-correspondences , planar case with known plane depth

1. Capture new frame img_{k+1}
2. Extract and match features between img_{k+1} and img_k
3. Estimate homography $H_{k,k+1}$
4. Decompose the $H_{k,k+1}$ and eliminate 3 out of the 4 possible solutions to get the relative pose
$$^{k}\xi_{k+1} = [^{k}R_{k+1} \quad ^{k}\boldsymbol{t}_{k+1}]$$
5. Calculate the pose of camera $k+1$ relative to the first camera
$$^{0}\xi_{k+1} = {^{0}\xi_{k}}\,{^{k}\xi_{k+1}}$$





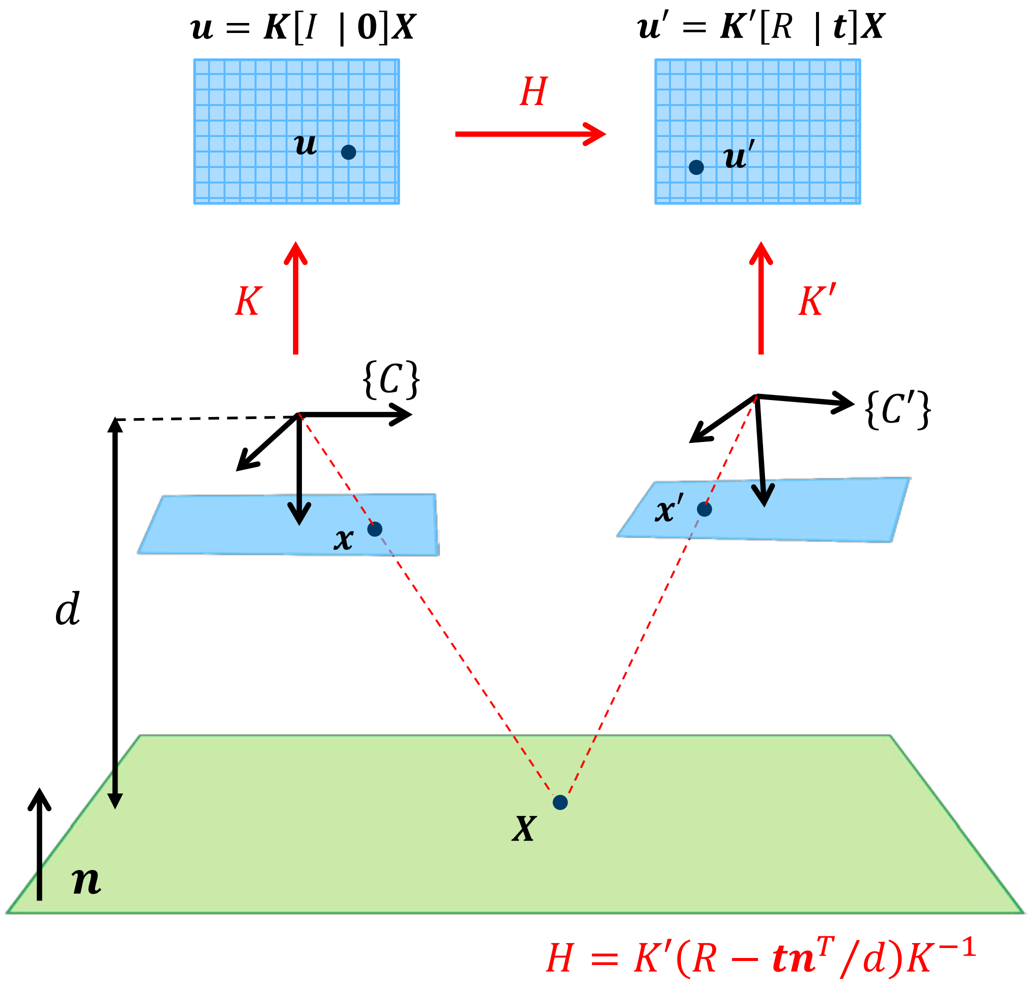
Planar scene

- This method is well suited for environments where we can detect scene planes and have some knowledge about the orientation of these planes

- Indoors
  - walls, floor, ceiling

- In city environments
  - ground, sides of buildings

- Imaging from high altitudes
  - ground



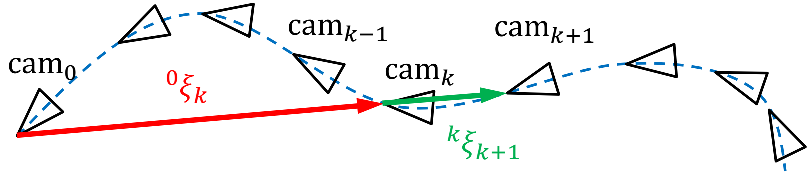
Summary

- Pose from epipolar geometry

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

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

- Visual odometry



- Additional reading:
  - Szeliski: 7.2

- Optional reading:
  - H. C Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, 1981
  - Ezio Malis & Manuel Vargas, *Deeper understanding of the homography decomposition for vision-based control, 2007*
  - Davide Scaramuzza, *Tutorial on Visual Odometry*