UiO : Department of Technology Systems
University of Oslo

Pose from epipolar geometry

Thomas Opsahl

2023

Introduction

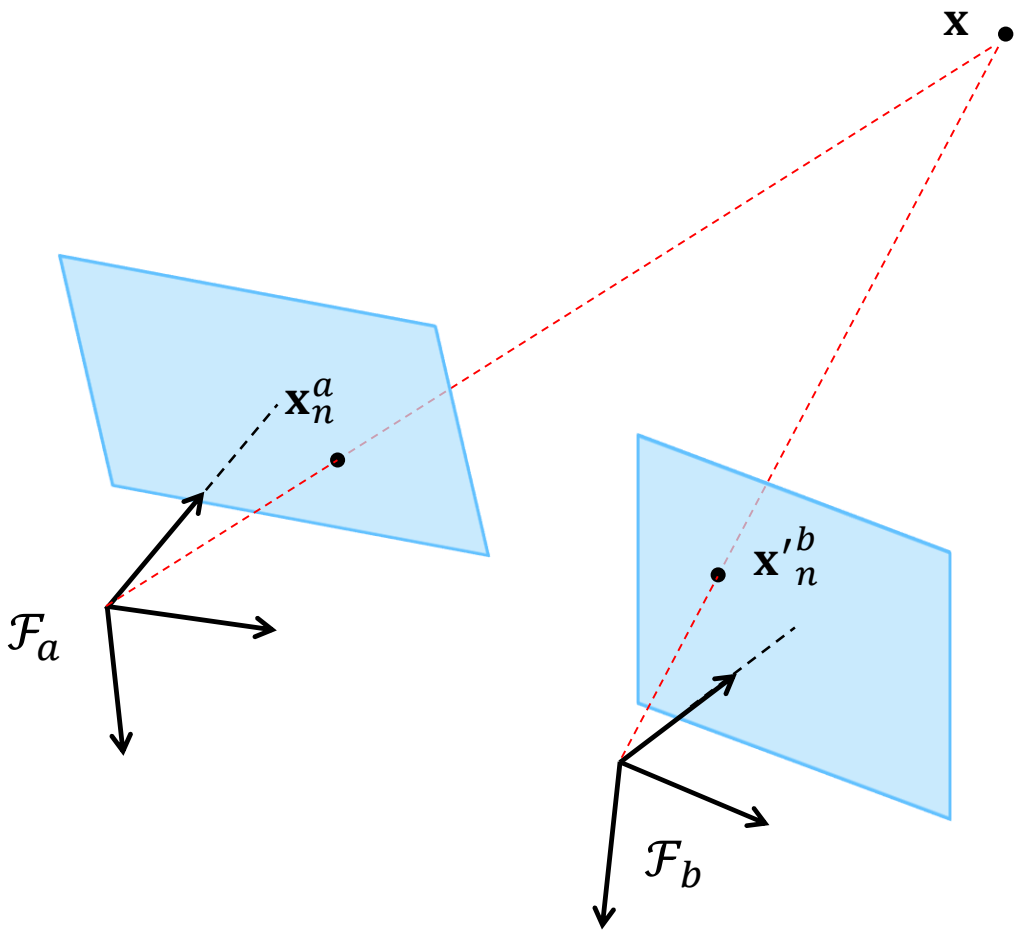
The essential matrix $\mathbf{E}$ represents the epipolar constraint on normalized image points $\mathbf{x}_n^a$ and $\mathbf{x}'^b_n$ corresponding to the same 3D point $\mathbf{x}$

$$\tilde{\mathbf{x}}'^{b\,T}_n \mathbf{E}_{ba} \tilde{\mathbf{x}}^a_n = 0$$

The essential matrix is closely related to the relative pose between the cameras

$$\mathbf{E}_{ba} = [\mathbf{t}^b_{ba}]_\times \mathbf{R}_{ba}$$

But $\mathbf{E}$ is homogeneous by nature, so the equality is only up to scale

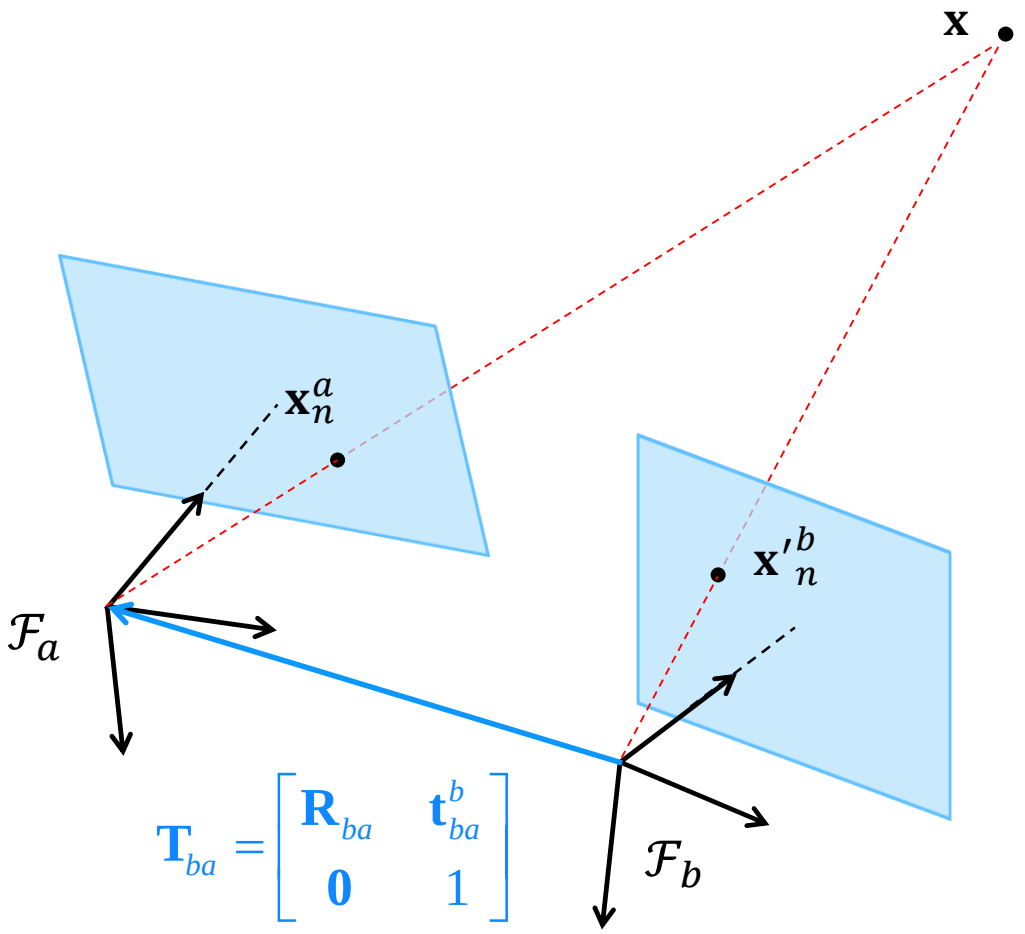


Introduction

The essential matrix has five degrees of freedom and can be estimated from five or more point correspondences $\mathbf{x}_n^a \leftrightarrow \mathbf{x}'^b_n$

One of the most important results in computer vision is that for a given essential matrix it is possible to determine the relative pose between the cameras up the scale of the translation vector

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



Pose from epipolar geometry

One of the properties of $\mathbf{E}_{ba}$ is that it only has two nonzero singular values that always are equal

This means that we can force these singular values to be 1 by rescaling $\mathbf{E}_{ba}$

Then its singular value decomposition has the form

$$\mathbf{E}_{ba} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \mathbf{v}_3^T \end{bmatrix}$$

Pose from epipolar geometry

One of the properties of $\mathbf{E}_{ba}$ is that it only has two nonzero singular values that always are equal

This means that we can force these singular values to be 1 by rescaling $\mathbf{E}_{ba}$

Then its singular value decomposition has the form

$$\mathbf{E}_{ba} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{v}_1^T\\ \mathbf{v}_2^T\\ \mathbf{v}_3^T\end{bmatrix}$$

There are four different poses that satisfy the equation $\mathbf{E}_{ba} = [\mathbf{t}_{ba}^b]_\times \mathbf{R}_{ba}$

$$\mathbf{T}_{ba,1} = \begin{bmatrix}\mathbf{U}\mathbf{W}\mathbf{V}^T & \mathbf{u}_3\\ \mathbf{0} & 1\end{bmatrix} \qquad \mathbf{T}_{ba,2} = \begin{bmatrix}\mathbf{U}\mathbf{W}\mathbf{V}^T & -\mathbf{u}_3\\ \mathbf{0} & 1\end{bmatrix}$$

$$\mathbf{T}_{ba,3} = \begin{bmatrix}\mathbf{U}\mathbf{W}^T\mathbf{V}^T & \mathbf{u}_3\\ \mathbf{0} & 1\end{bmatrix} \qquad \mathbf{T}_{ba,4} = \begin{bmatrix}\mathbf{U}\mathbf{W}^T\mathbf{V}^T & -\mathbf{u}_3\\ \mathbf{0} & 1\end{bmatrix}$$

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

Pose from epipolar geometry

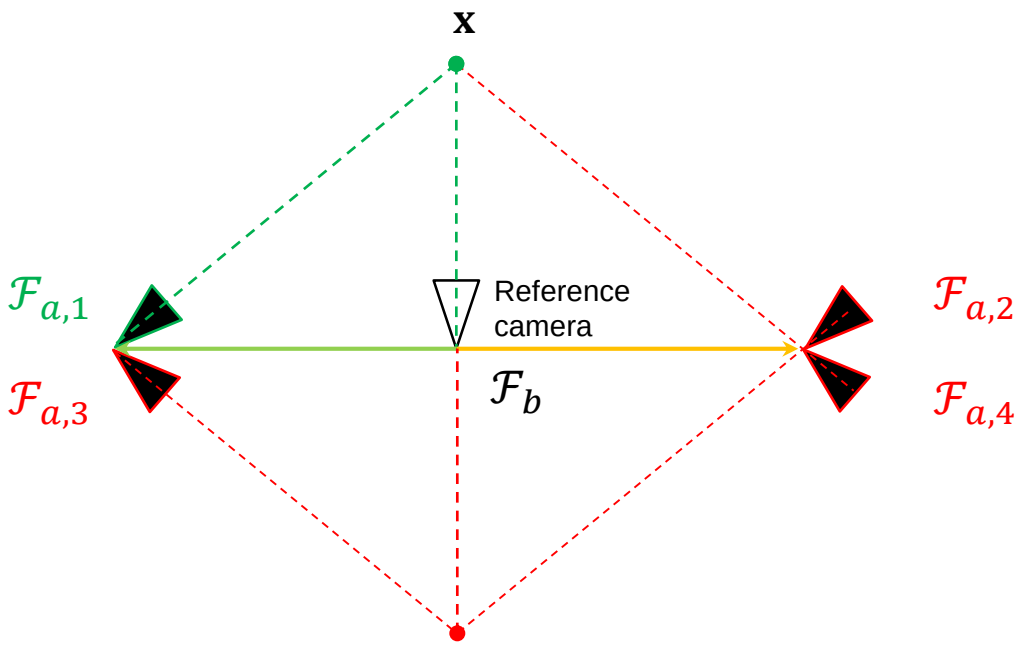
The reason for it being four solutions is that the perspective camera model does not differentiate between points in front of the camera and points that are behind it

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

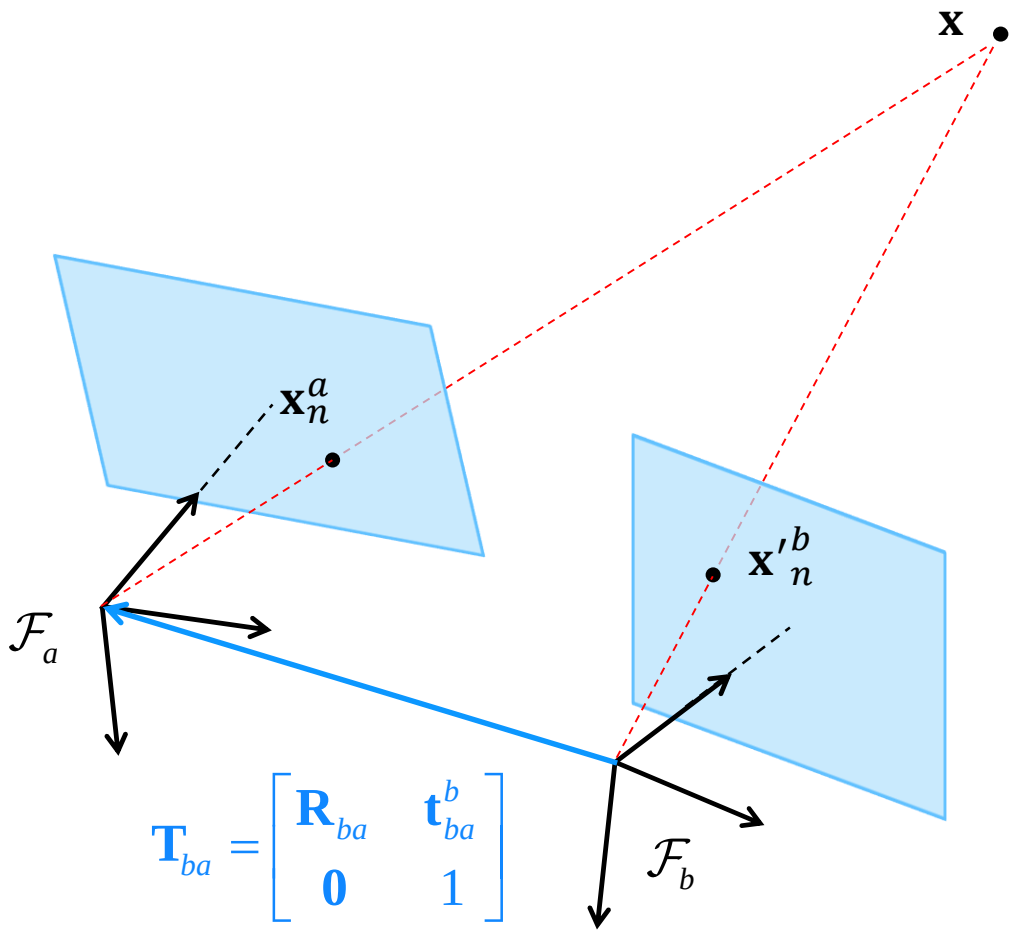
This constraint is known as **the chirality constraint** and it is tested by triangulation of at least one 3D point



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 correspondences $\mathbf{x}_{n,i}^a \leftrightarrow \mathbf{x}'^b_{n,i}$ 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 $\mathbf{T}_{ba,1}, \ldots, \mathbf{T}_{ba,4}$ 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

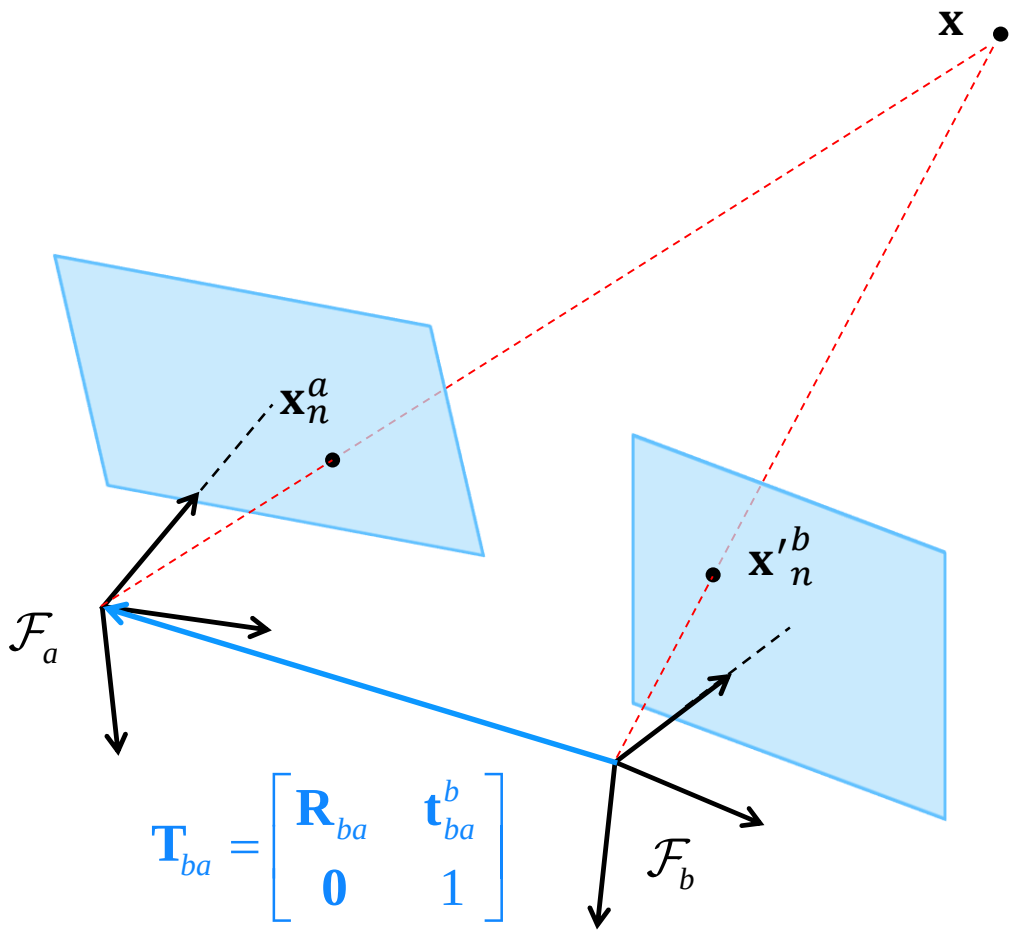


Pose from epipolar geometry

Pose between two calibrated cameras

1. Establish robust correspondences $\mathbf{u}_i^a \leftrightarrow \mathbf{u}'^b_i$ between images
2. Determine correspondences $\mathbf{x}_{n,i}^a \leftrightarrow \mathbf{x}'^b_{n,i}$ 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 $\mathbf{T}_{ba,1}, \ldots, \mathbf{T}_{ba,4}$ 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

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



Visual odometry – A naïve approach

Based on what we now know it is possible to
use a camera for relative navigation

This is called **visual odometry** or just **VO**

Visual odometry – A naïve approach

Based on what we now know it is possible to use a camera for relative navigation

This is called **visual odometry** or just **VO**

A naïve algorithm can look something like this

1. Capture two frames img_0 and img_1 and establish correspondences $\mathbf{x}^0_{n,i} \leftrightarrow \mathbf{x}'^1_{n,i}$
2. Estimate $\mathbf{E}_{1\,0}$ and determine the pose of $\mathcal{F}_1$ relative to $\mathcal{F}_0$
3. Determine the 3D point cloud $\{\mathbf{x}_i\}$ by triangulating all correspondences $\mathbf{x}^0_{n,i} \leftrightarrow \mathbf{x}'^1_{n,i}$
4. Capture img_2 and establish correspondences $\mathbf{x}^1_{n,i} \leftrightarrow \mathbf{x}'^2_{n,i}$
5. This yields 3D-2D correspondences $\mathbf{x}^2_{n,i} \leftrightarrow \mathbf{x}_i$ between img_2 and the point cloud $\{\mathbf{x}_i\}$
6. Estimate the pose of $\mathcal{F}_2$ from 3D-2D correspondences $\mathbf{x}^2_{n,i} \leftrightarrow \mathbf{x}_i$

Visual odometry – A naïve approach

Based on what we now know it is possible to use a camera for relative navigation

This is called **visual odometry** or just **VO**

A naïve algorithm can look something like this

repeat

1. Capture two frames img_0 and img_1 and establish correspondences $\mathbf{x}^0_{n,i} \leftrightarrow \mathbf{x}'^1_{n,i}$
2. Estimate $\mathbf{E}_{1\,0}$ and determine the pose of $\mathcal{F}_1$ relative to $\mathcal{F}_0$
3. Determine the 3D point cloud $\{\mathbf{x}_i\}$ by triangulating all correspondences $\mathbf{x}^0_{n,i} \leftrightarrow \mathbf{x}'^1_{n,i}$
4. Capture img_2 and establish correspondences $\mathbf{x}^1_{n,i} \leftrightarrow \mathbf{x}'^2_{n,i}$
5. This yields 3D-2D correspondences $\mathbf{x}^2_{n,i} \leftrightarrow \mathbf{x}_i$ between img_2 and the point cloud $\{\mathbf{x}_i\}$
6. Estimate the pose of $\mathcal{F}_2$ from 3D-2D correspondences $\mathbf{x}^2_{n,i} \leftrightarrow \mathbf{x}_i$

Visual odometry – A naïve approach

It is common to let the first iteration define the scale by setting $\|\mathbf{t}_{1\,0}\| = 1$

The 3D point cloud will then have the same scale and "pass it on" to the next camera pose

Due to noise in measured and computed quantities it is difficult to preserve the scale over many iterations

This is known as **scale drift**

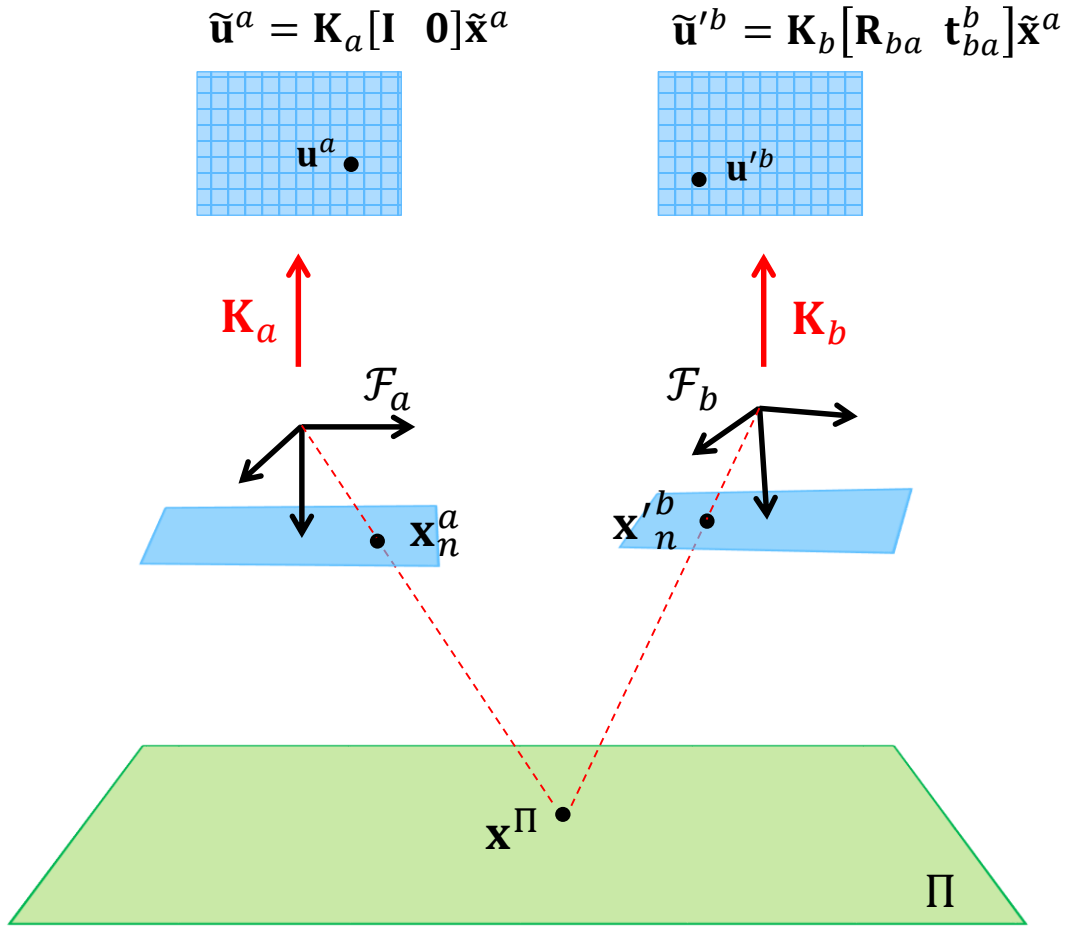
A stereo camera has a built-in scale, the baseline, and is thus better suited for VO than a single camera

1. Capture two frames img_0 and img_1 and establish correspondences $\mathbf{x}_{n,i}^{0} \leftrightarrow \mathbf{x}'^{1}_{n,i}$
2. Estimate $\mathbf{E}_{1\,0}$ and determine the pose of $\mathcal{F}_1$ relative to $\mathcal{F}_0$
3. Determine the 3D point cloud $\{\mathbf{x}_i\}$ by triangulating all correspondences $\mathbf{x}_{n,i}^{0} \leftrightarrow \mathbf{x}'^{1}_{n,i}$
4. Capture img_2 and establish correspondences $\mathbf{x}_{n,i}^{1} \leftrightarrow \mathbf{x}'^{2}_{n,i}$
5. This yields 3D-2D correspondences $\mathbf{x}_{n,i}^{2} \leftrightarrow \mathbf{x}_i$ between img_2 and the point cloud $\{\mathbf{x}_i\}$
6. Estimate the pose of $\mathcal{F}_2$ from 3D-2D correspondences $\mathbf{x}_{n,i}^{2} \leftrightarrow \mathbf{x}_i$

Planar scene

For planar scenes, it is not possible to estimate the epipolar geometry, $\mathbf{E}$ or $\mathbf{F}$, from correspondences

- For almost planar scenes, the estimation is likely to be ill-conditioned

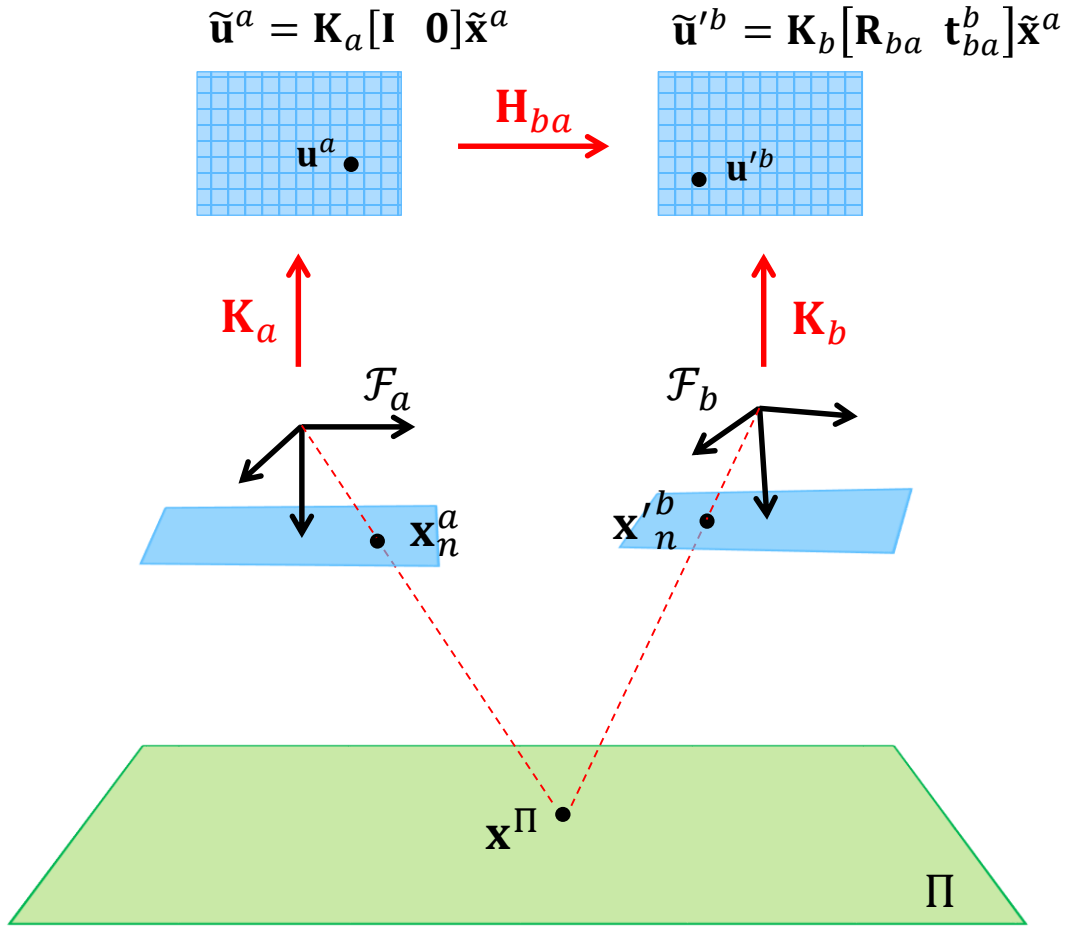


Planar scene

For planar scenes, it is not possible to estimate the epipolar geometry, $\mathbf{E}$ or $\mathbf{F}$, from correspondences

- For almost planar scenes, the estimation is likely to be ill-conditioned

Two images of a planar scene are however related by a homography $\mathbf{H}$



Planar scene

For planar scenes, it is not possible to estimate the epipolar geometry, $\mathbf{E}$ or $\mathbf{F}$, from correspondences

- For almost planar scenes, the estimation is likely to be ill-conditioned

Two images of a planar scene are however related by a homography $\mathbf{H}$

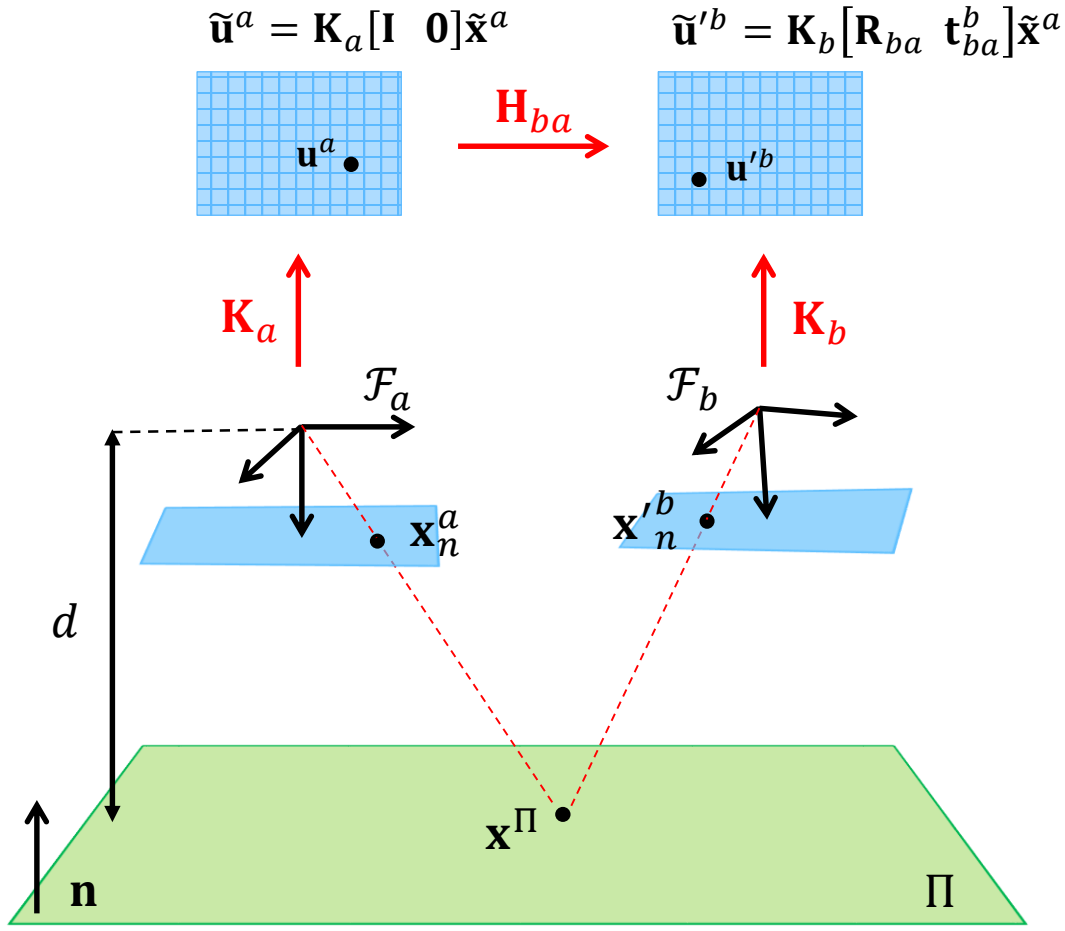
One can prove that if

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

then

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

where $\mathbf{n}$ is the normal vector of the plane and d is the distance of the plane relative to $\mathcal{F}_a$

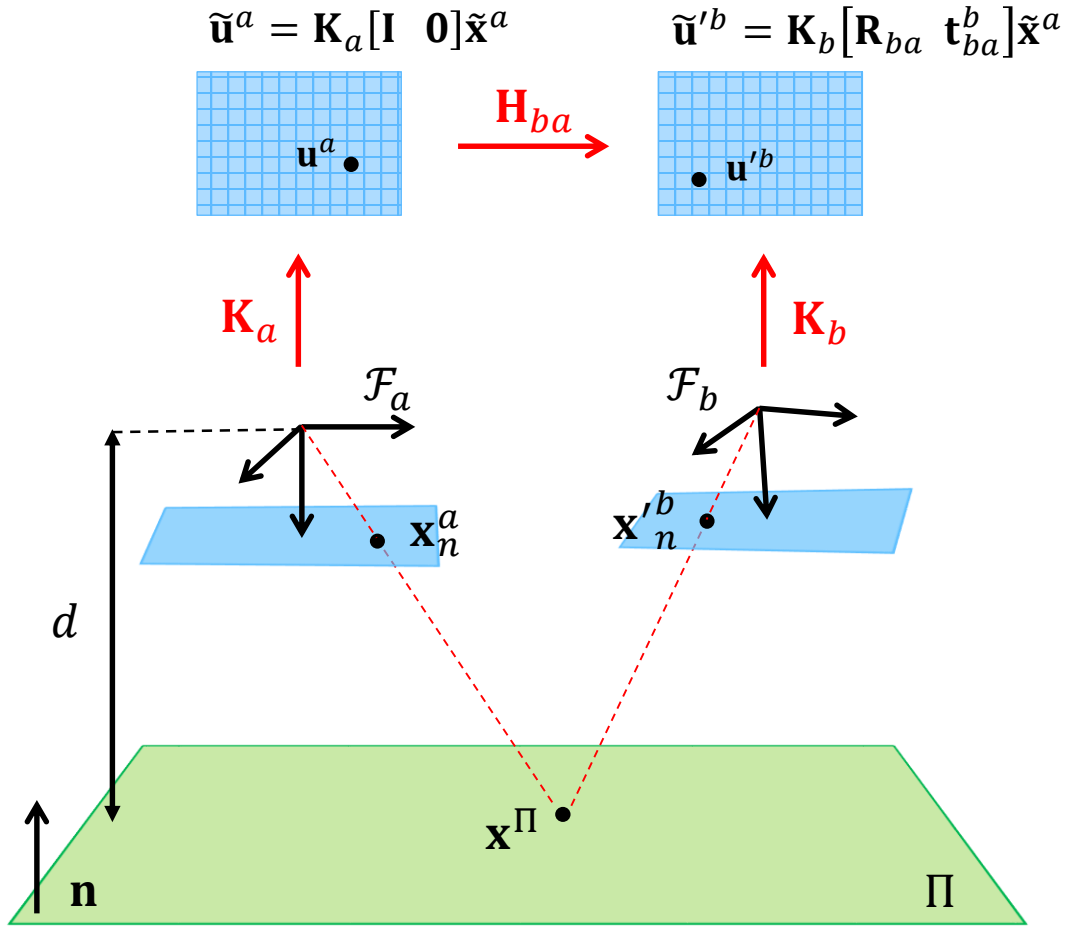


Planar scene

Based on the expression for $\mathbf{H}_{ba}$ it is possible to estimate $\left(\mathbf{R}_{ba}, \mathbf{n}^a, \frac{1}{d}\mathbf{t}_{ba}^b\right)$ from a known homography

This is known as *homography decomposition* and it has been shown to have four solutions

- Two solutions can be invalidated by requiring points to be in front of both cameras
- With some prior knowledge about the geometry, e.g. the normal vector $\mathbf{n}$, it is often possible to eliminate one of the two remaining solutions



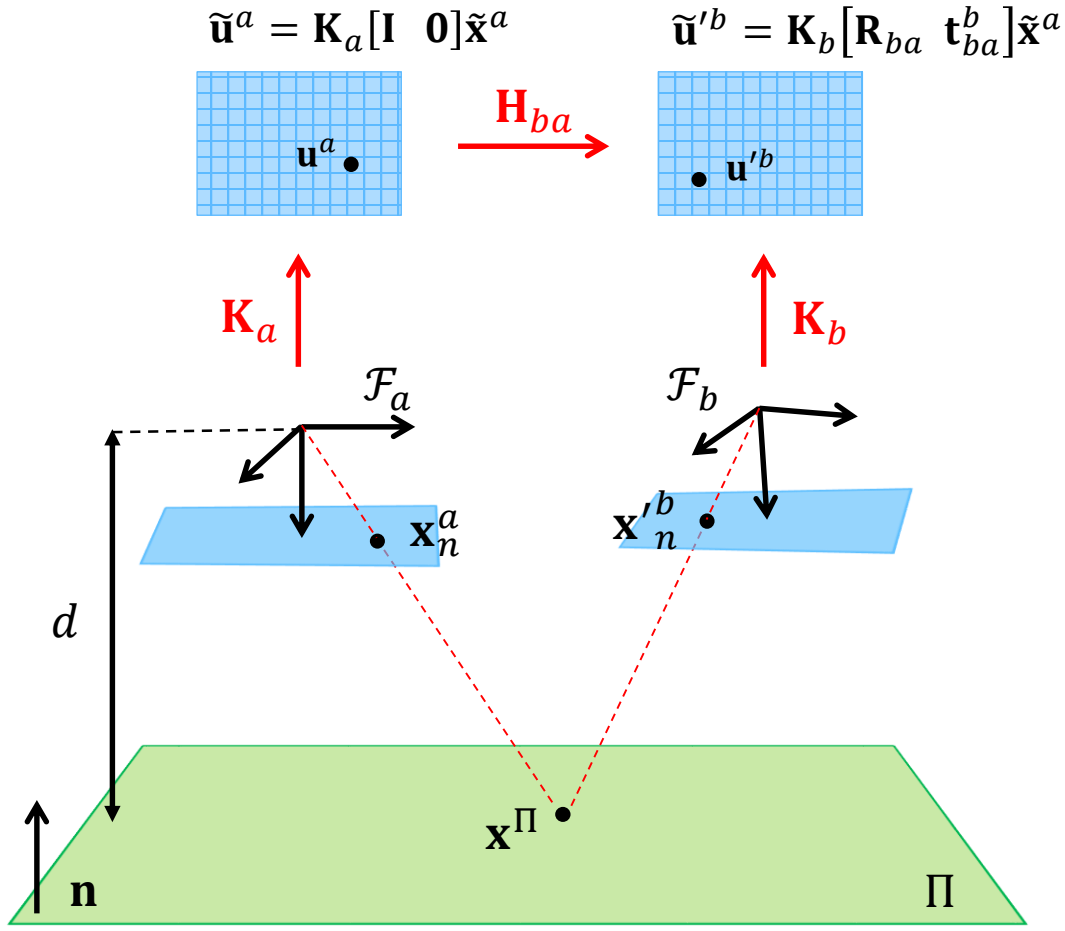
$$\mathbf{H}_{ba} = \mathbf{K}_b\left(\mathbf{R}_{ba} - \mathbf{t}_{ba}^b(\mathbf{n}^a)^T/d\right)\mathbf{K}_a^{-1}$$

Planar scene

This can be used for VO in cases where the scene contains planar surfaces that we can detect and know the orientation of

- Indoors
  - walls, floor, ceiling
- In city environments
  - ground, sides of buildings
- Imaging from high altitudes
  - ground

Then we should be able to reject 3 of the 4 poses provided by homography decomposition



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

Summary

Pose from epipolar geometry

From the essential matrix $\mathbf{E}_{ba}$ we can estimate the relative pose $\mathbf{T}_{ba}$ between two cameras $\mathcal{F}_a$ and $\mathcal{F}_b$

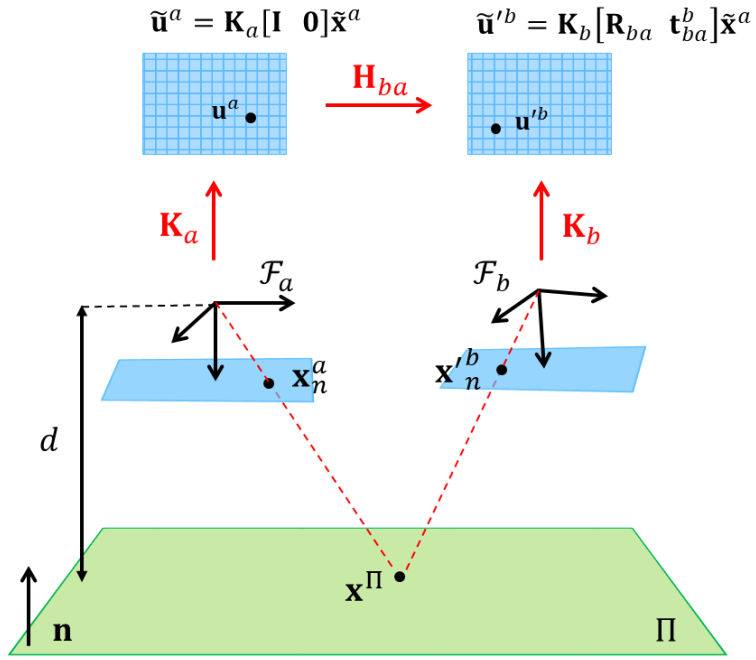
$$\mathbf{E}_{ba} \rightarrow \mathbf{T}_{ba} = \begin{bmatrix} \mathbf{R}_{ba} & \mathbf{t}_{ba}^b \\ \mathbf{0} & 1 \end{bmatrix}$$

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

Visual Odometry

We now know methods that can be important components in a VO algorithm

- $\mathbf{u}_i^a \leftrightarrow \mathbf{u}'^b_i \rightarrow \mathbf{T}_{ba}$
- $\mathbf{u}_i^a \leftrightarrow \mathbf{u}'^b_i \rightarrow \mathbf{x}_i$
- $\mathbf{x}_i^a \leftrightarrow \mathbf{u}_i^b \rightarrow \mathbf{T}_{ba}$



Planar scenes

From the homography $\mathbf{H}_{ba}$ we can estimate $\left(\mathbf{R}_{ba}, \mathbf{n}^a, \frac{1}{d}\mathbf{t}_{ba}^b\right)$ for two cameras $\mathcal{F}_a$ and $\mathcal{F}_b$, but it requires some knowledge about the geometry

Supplementary material

Recommended

- *Richard Szeliski: Computer Vision: Algorithms and Applications 2nd ed*
  - Chapter 11 “Structure from motion and SLAM”, in particular section 11.2 “Pose estimation”

Other

- Davide Scaramuzza, *Tutorial on Visual Odometry*, 2012

- Ezio Malis & Manuel Vargas, *Deeper understanding of the homography decomposition for vision-based control*, 2007