Lecture 7.2
Triangulation

Thomas Opsahl

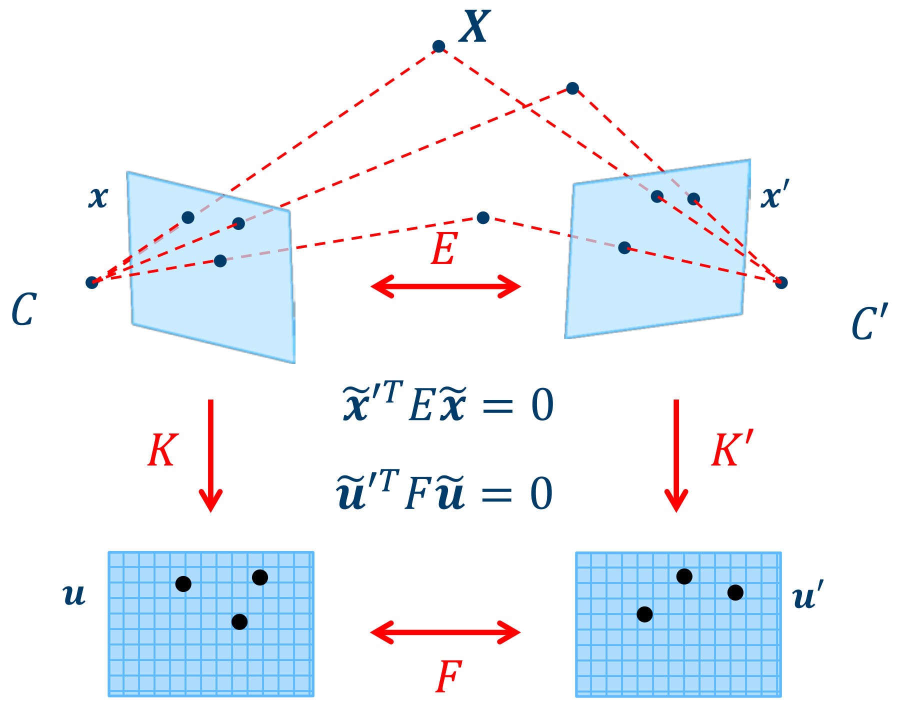
Introduction

- We have seen that two perspective cameras observing the same points must satisfy the epipolar constraint

- The essential matrix $E = [\boldsymbol{t}]_{\times} R$
$$\widetilde{\boldsymbol{x}}'^T E \widetilde{\boldsymbol{x}} = 0$$

- The fundamental matrix $F = K'^{-T} E K^{-1}$
$$\widetilde{\boldsymbol{u}}'^T F \widetilde{\boldsymbol{u}} = 0$$

- Being observed by two perspective cameras also puts a strong geometric constraint on the observed points $\boldsymbol{X}_i$
$$P\boldsymbol{X}_i = \boldsymbol{u}_i$$
$$P'\boldsymbol{X}_i = \boldsymbol{u}_i'$$



- In the following we will look at how we can estimate 3D points $\boldsymbol{X}_i$ from known camera matrices P, P' and 2D correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}_i'$

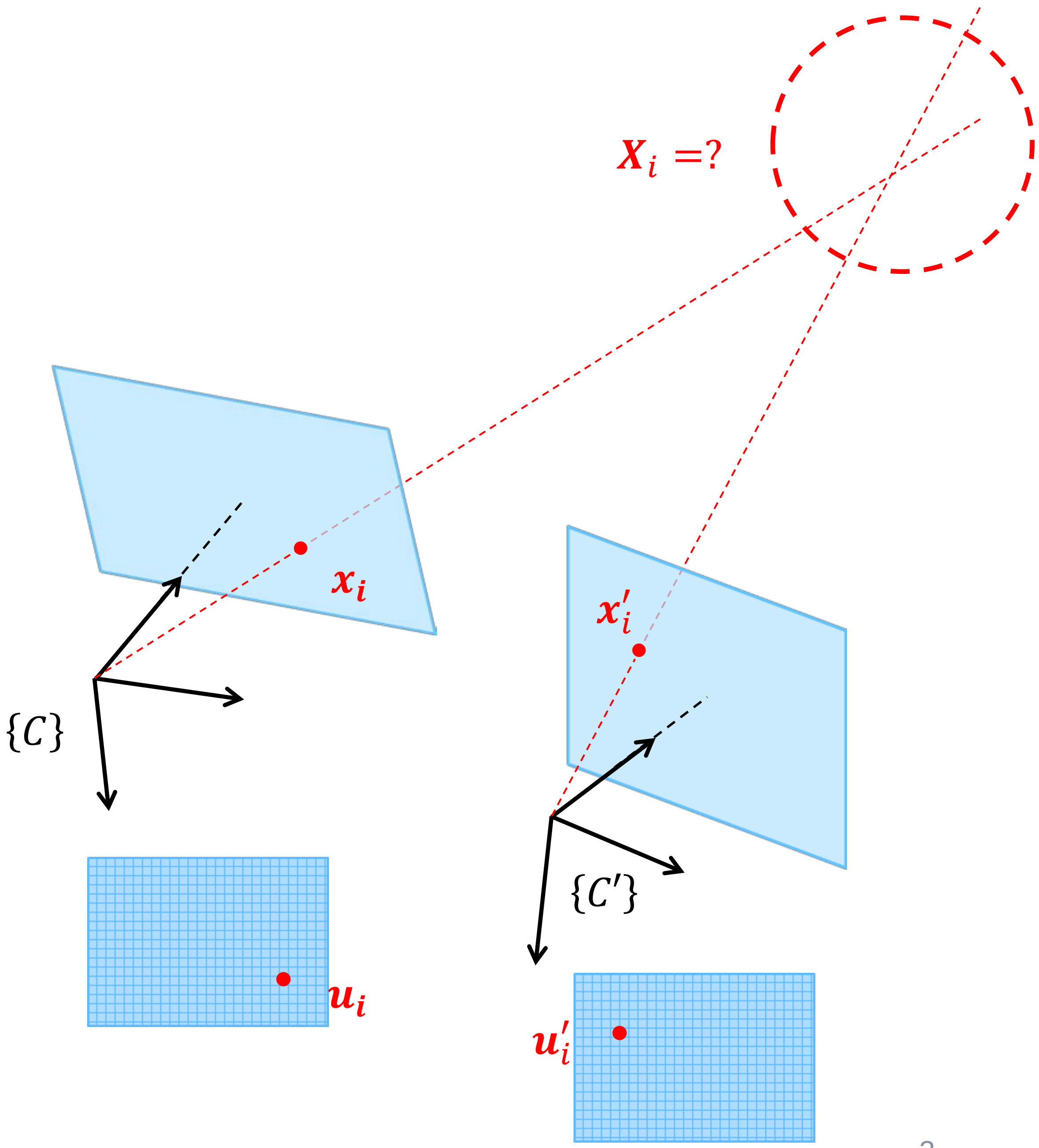
Introduction

- Assume that we know the camera matrices P, P' and 2D correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}_i'$

- In order to determine the 3D point $\boldsymbol{X}_i$ it is tempting to back-project the two image points and determine their intersection

- But due to noise, the two rays in 3D will "never" truly intersect, so we need to estimate a best solution to the problem

- Several ways to approach the problem depending on what we choose to optimize over
  - Only errors in $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}_i'$?
  - Errors in $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}_i'$, P and P'?

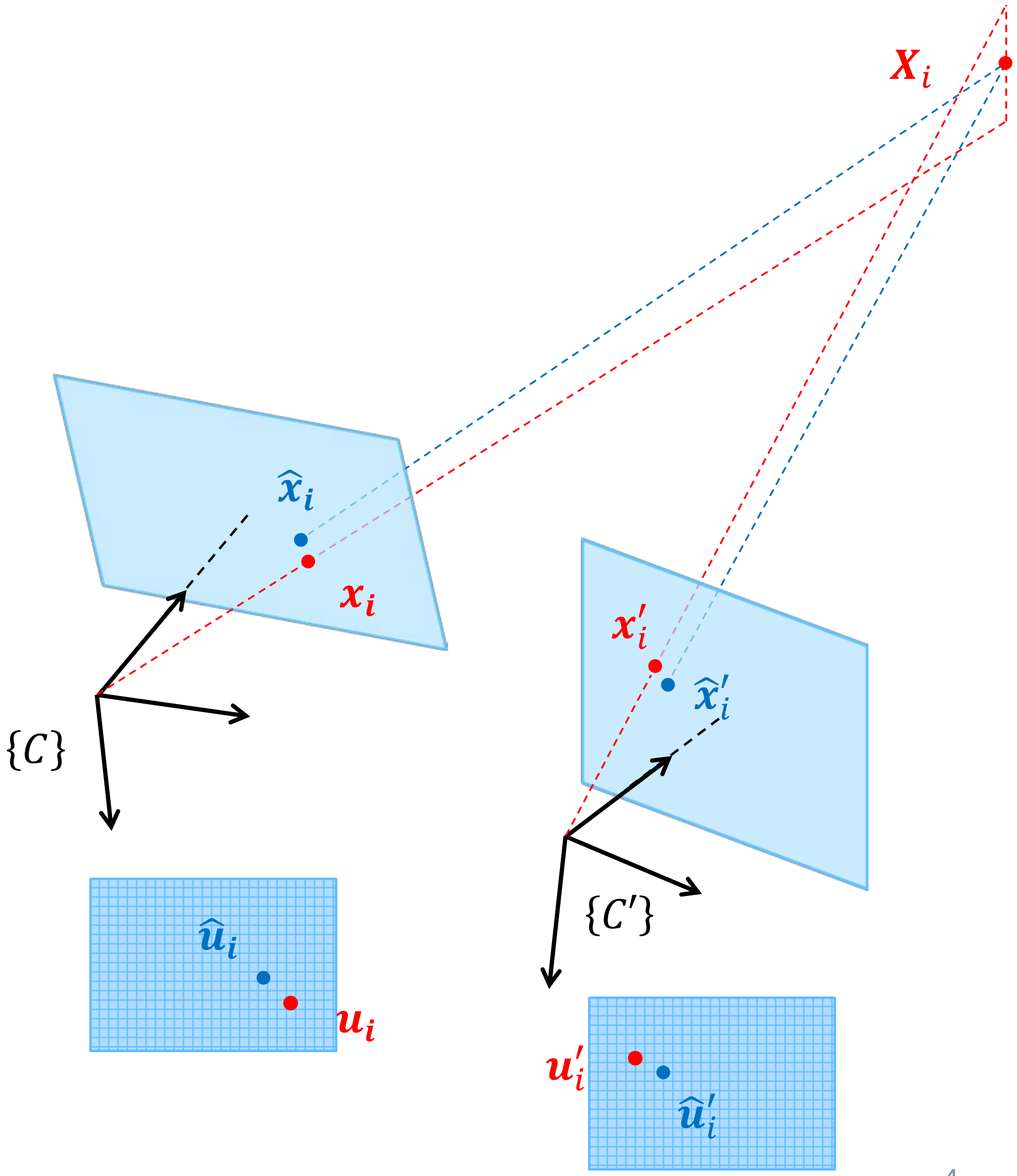


The 3D mid-point
Minimizing the 3D error

- One natural estimate for $\boldsymbol{X}_i$ is the midpoint on the shortest line between to two back-projected rays

- This minimize the 3D error, but typically not the reprojection error

$$\epsilon_i = d(\boldsymbol{u}_i, P\boldsymbol{X}_i)^2 + d(\boldsymbol{u}_i', P'\boldsymbol{X}_i)^2$$
$$\epsilon_i = d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2 + d(\boldsymbol{u}_i', \hat{\boldsymbol{u}}_i')^2$$

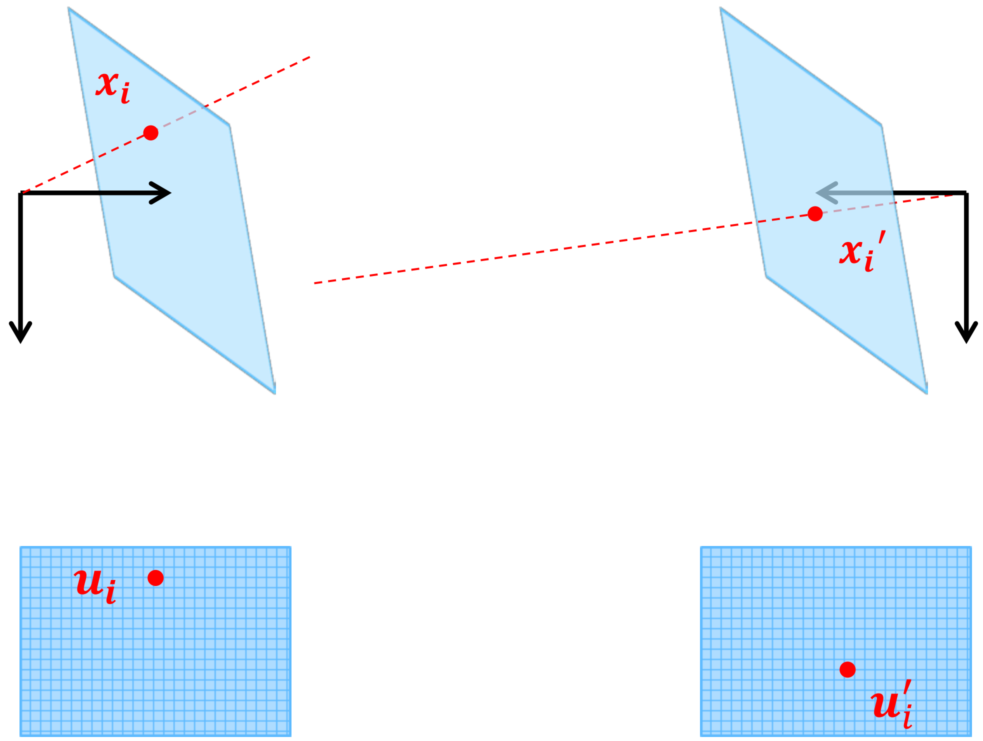


The 3D mid-point
Minimizing the 3D error

- One natural estimate for $\boldsymbol{X}_i$ is the midpoint on the shortest line between to two back-projected rays

- This minimize the 3D error, but typically not the reprojection error
$$\epsilon_i = d(\boldsymbol{u}_i, P\boldsymbol{X}_i)^2 + d(\boldsymbol{u}_i', P'\boldsymbol{X}_i)^2$$
$$\epsilon_i = d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2 + d(\boldsymbol{u}_i', \hat{\boldsymbol{u}}_i')^2$$

- The difference becomes clear when $\boldsymbol{X}_i$ is much closer to one of the cameras than the other



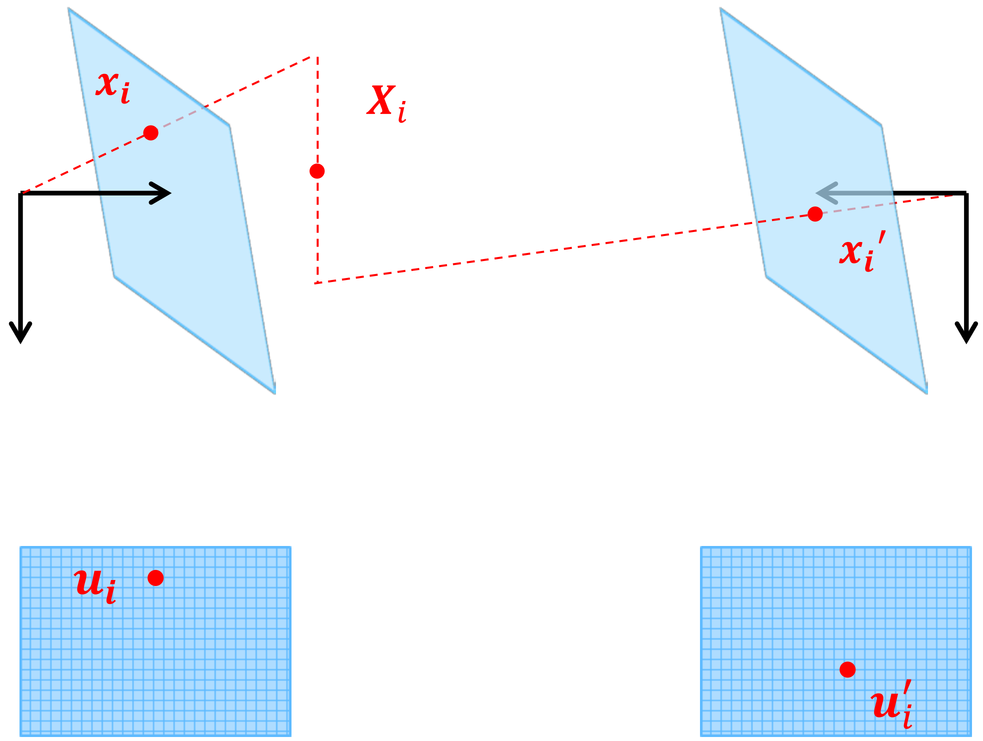
The 3D mid-point
Minimizing the 3D error

- One natural estimate for $\boldsymbol{X}_i$ is the midpoint on the shortest line between to two back-projected rays

- This minimize the 3D error, but typically not the reprojection error

$$\epsilon_i = d(\boldsymbol{u}_i, P\boldsymbol{X}_i)^2 + d(\boldsymbol{u}_i', P'\boldsymbol{X}_i)^2$$
$$\epsilon_i = d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2 + d(\boldsymbol{u}_i', \hat{\boldsymbol{u}}_i')^2$$

- The difference becomes clear when $\boldsymbol{X}_i$ is much closer to one of the cameras than the other



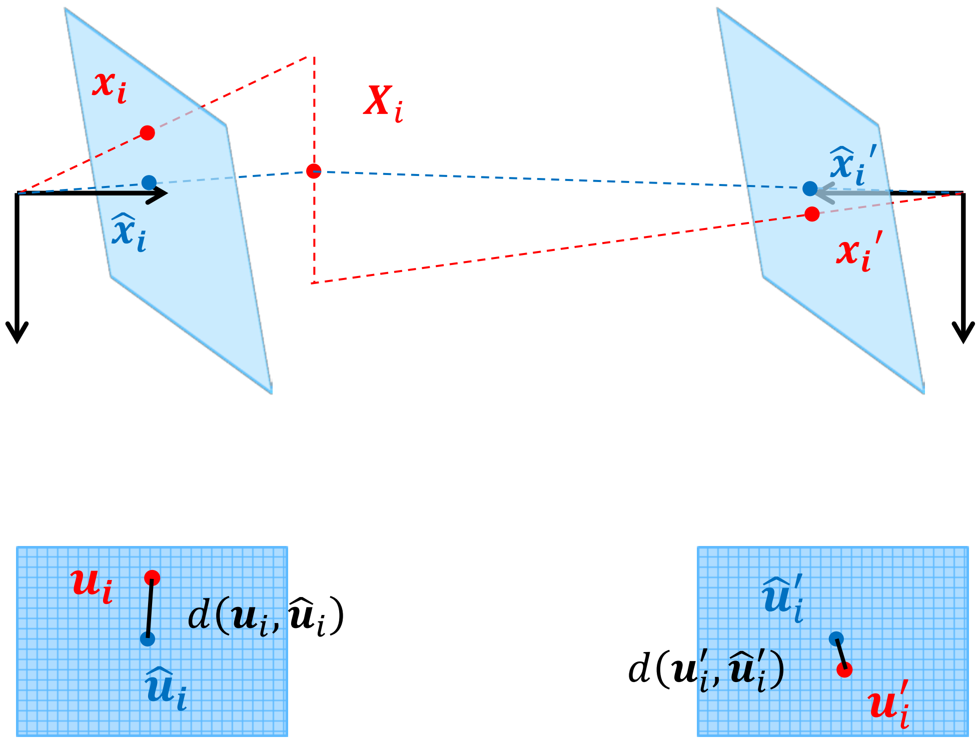
The 3D mid-point
Minimizing the 3D error

- One natural estimate for $\boldsymbol{X}_i$ is the midpoint on the shortest line between to two back-projected rays

- This minimize the 3D error, but typically not the reprojection error

$$\epsilon_i = d(\boldsymbol{u}_i, P\boldsymbol{X}_i)^2 + d(\boldsymbol{u}_i', P'\boldsymbol{X}_i)^2$$
$$\epsilon_i = d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2 + d(\boldsymbol{u}_i', \hat{\boldsymbol{u}}_i')^2$$

- The difference becomes clear when $\boldsymbol{X}_i$ is much closer to one of the cameras than the other



The 3D mid-point

Minimizing the 3D error

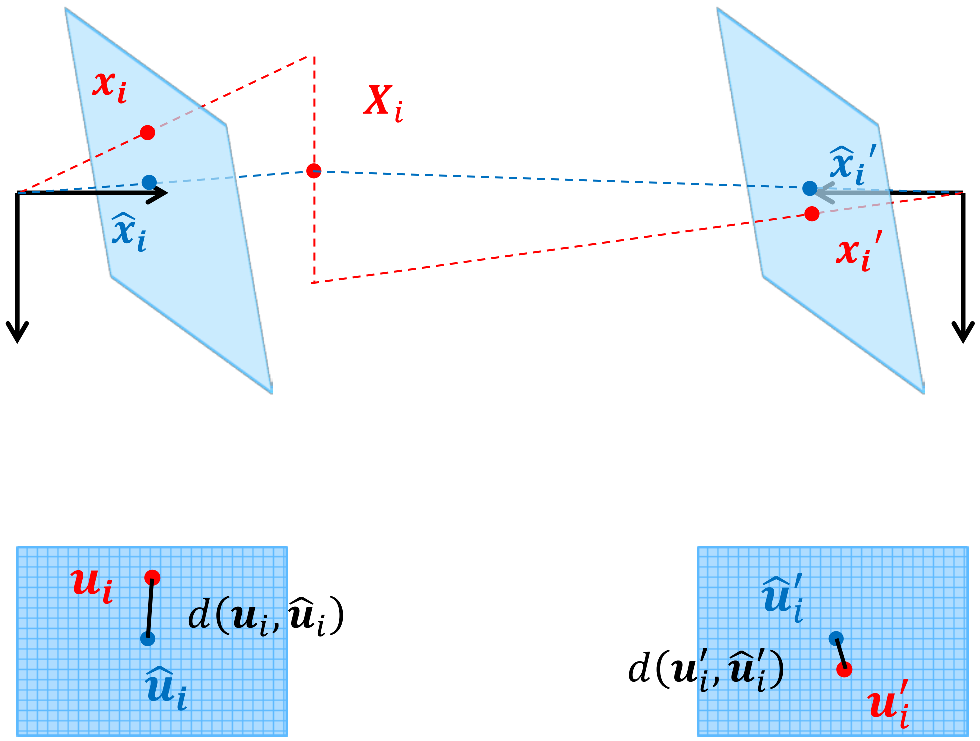
- One natural estimate for $\boldsymbol{X}_i$ is the midpoint on the shortest line between to two back-projected rays

- This minimize the 3D error, but typically not the reprojection error

$$\epsilon_i = d(\boldsymbol{u}_i, P\boldsymbol{X}_i)^2 + d(\boldsymbol{u}'_i, P'\boldsymbol{X}_i)^2$$
$$\epsilon_i = d(\boldsymbol{u}_i, \hat{\boldsymbol{u}}_i)^2 + d(\boldsymbol{u}'_i, \hat{\boldsymbol{u}}'_i)^2$$

- The difference becomes clear when $\boldsymbol{X}_i$ is much closer to one of the cameras than the other

- Another disadvantage of this method is that it does not extend naturally to situations when $\boldsymbol{X}_i$ is observed by more than two cameras



Linear triangulation

Minimizing the algebraic error

- This algorithm uses the two equations for perspective projection to solve for the 3D point that are optimal in a least squares sense

- Each perspective camera model gives rise to two equations on the three entries of $\boldsymbol{X}_i$

$$\tilde{\boldsymbol{u}}_i = P\tilde{\boldsymbol{X}}_i \qquad\qquad \tilde{\boldsymbol{u}}'_i = P'\tilde{\boldsymbol{X}}_i$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$\tilde{\boldsymbol{u}}_i \times P\tilde{\boldsymbol{X}}_i = \boldsymbol{0} \qquad\qquad \tilde{\boldsymbol{u}}'_i \times P'\tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{p}^{1T} \\ \boldsymbol{p}^{2T} \\ \boldsymbol{p}^{3T} \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0} \qquad\qquad \begin{bmatrix} u'_i \\ v'_i \\ 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{p}'^{1T} \\ \boldsymbol{p}'^{2T} \\ \boldsymbol{p}'^{3T} \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

$$\begin{bmatrix} v_i \boldsymbol{p}^{3T} - \boldsymbol{p}^{2T} \\ \boldsymbol{p}^{1T} - u_i \boldsymbol{p}^{3T} \\ u_i \boldsymbol{p}^{2T} - v_i \boldsymbol{p}^{1T} \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0} \qquad\qquad \begin{bmatrix} v'_i \boldsymbol{p}'^{3T} - \boldsymbol{p}'^{2T} \\ \boldsymbol{p}'^{1T} - u'_i \boldsymbol{p}'^{3T} \\ u'_i \boldsymbol{p}'^{2T} - v'_i \boldsymbol{p}'^{1T} \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

$$\Updownarrow \qquad\qquad \Updownarrow$$

$$\begin{bmatrix} v_i \boldsymbol{p}^{3T} - \boldsymbol{p}^{2T} \\ u_i \boldsymbol{p}^{3T} - \boldsymbol{p}^{1T} \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0} \qquad\qquad \begin{bmatrix} v'_i \boldsymbol{p}'^{3T} - \boldsymbol{p}'^{2T} \\ u'_i \boldsymbol{p}'^{3T} - \boldsymbol{p}'^{1T} \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

Linear triangulation

Minimizing the algebraic error

- This algorithm uses the two equations for perspective projection to solve for the 3D point that are optimal in a least squares sense

- Each perspective camera model gives rise to two equations on the three entries of $\boldsymbol{X}_i$

- Combining these equations we get an over determined homogeneous system of linear equations that we can solve with SVD

$$\begin{bmatrix} v_i \boldsymbol{p}^{3T} - \boldsymbol{p}^{2T} \\ u_i \boldsymbol{p}^{3T} - \boldsymbol{p}^{1T} \\ v_i' \boldsymbol{p}'^{3T} - \boldsymbol{p}'^{2T} \\ u_i' \boldsymbol{p}'^{3T} - \boldsymbol{p}'^{1T} \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

$$\begin{bmatrix} v_i p_{31} - p_{21} & v_i p_{32} - p_{22} & v_i p_{33} - p_{23} & v_i p_{34} - p_{24} \\ u_i p_{31} - p_{11} & u_i p_{32} - p_{12} & u_i p_{33} - p_{13} & u_i p_{34} - p_{14} \\ v_i' p_{31}' - p_{21}' & v_i' p_{32}' - p_{22}' & v_i' p_{33}' - p_{23}' & v_i' p_{34}' - p_{24}' \\ u_i' p_{31}' - p_{11}' & u_i' p_{32}' - p_{12}' & u_i' p_{33}' - p_{13}' & u_i' p_{34}' - p_{14}' \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

$$A\tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

Linear triangulation
Minimizing the algebraic error

- This algorithm uses the two equations for perspective projection to solve for the 3D point that are optimal in a least squares sense

- Each perspective camera model gives rise to two equations on the three entries of $\boldsymbol{X}_i$

- Combining these equations we get an over determined homogeneous system of linear equations that we can solve with SVD

- The minimized algebraic error is not geometrically meaningful, but the method extends naturally to the case when $\boldsymbol{X}_i$ is observed in more than two images

$$\begin{bmatrix} v_i \boldsymbol{p}^{3T} - \boldsymbol{p}^{2T} \\ u_i \boldsymbol{p}^{3T} - \boldsymbol{p}^{1T} \\ v_i' \boldsymbol{p}'^{3T} - \boldsymbol{p}'^{2T} \\ u_i' \boldsymbol{p}'^{3T} - \boldsymbol{p}'^{1T} \\ \vdots \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

$$\begin{bmatrix} v_i p_{31} - p_{21} & v_i p_{32} - p_{22} & v_i p_{33} - p_{23} & v_i p_{34} - p_{24} \\ u_i p_{31} - p_{11} & u_i p_{32} - p_{12} & u_i p_{33} - p_{13} & u_i p_{34} - p_{14} \\ v_i' p_{31}' - p_{21}' & v_i' p_{32}' - p_{22}' & v_i' p_{33}' - p_{23}' & v_i' p_{34}' - p_{24}' \\ u_i' p_{31}' - p_{11}' & u_i' p_{32}' - p_{12}' & u_i' p_{33}' - p_{13}' & u_i' p_{34}' - p_{14}' \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

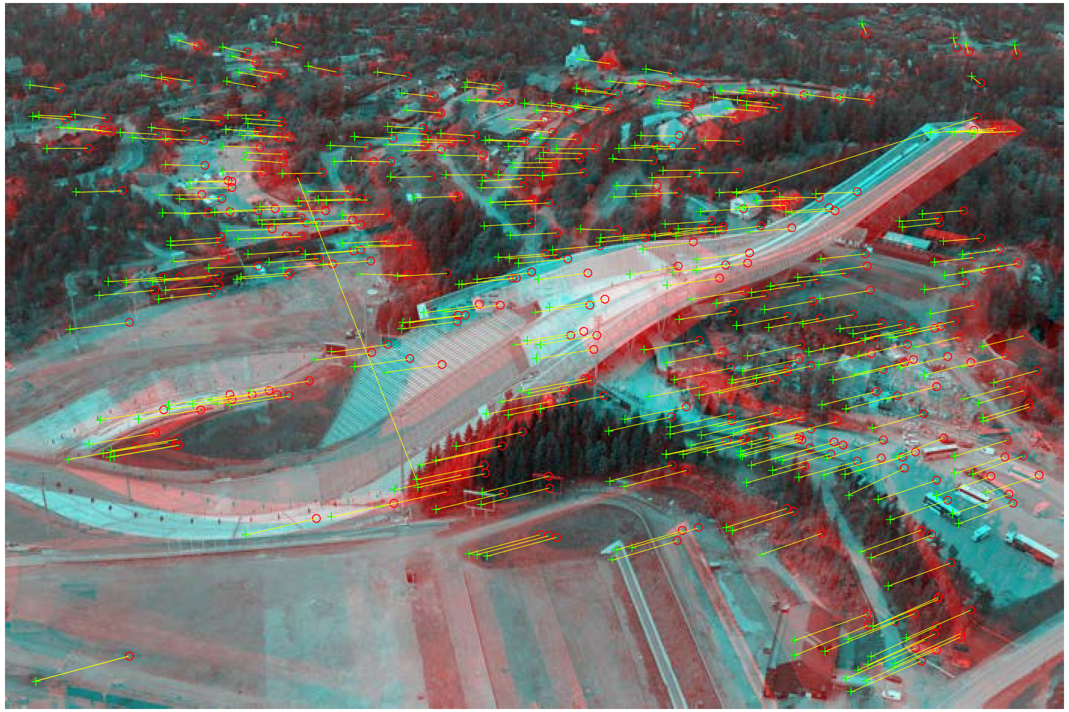
$$A\tilde{\boldsymbol{X}}_i = \boldsymbol{0}$$

Example

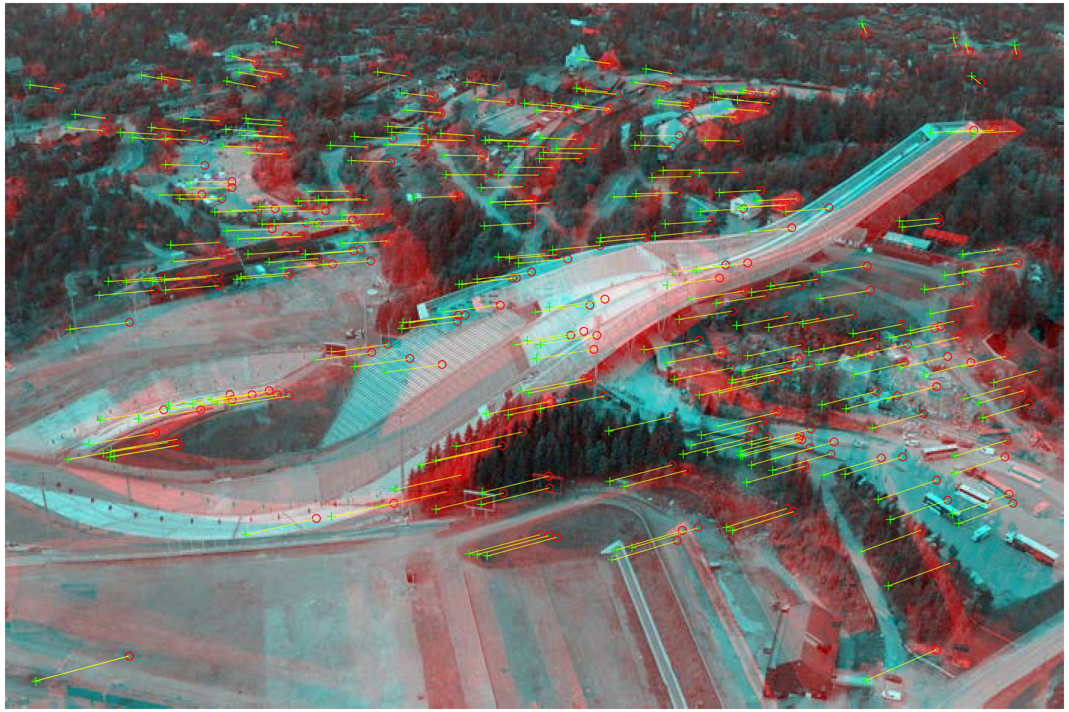
- Two views with known relative pose

Example



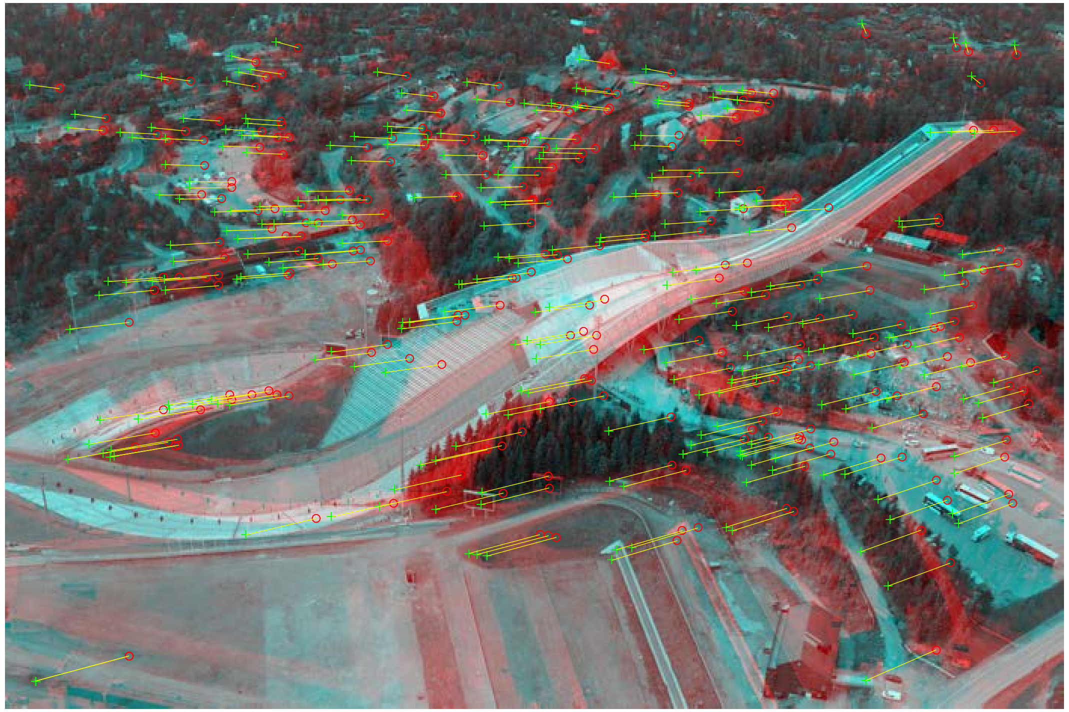
- Two views with known relative pose
- Matching feature points

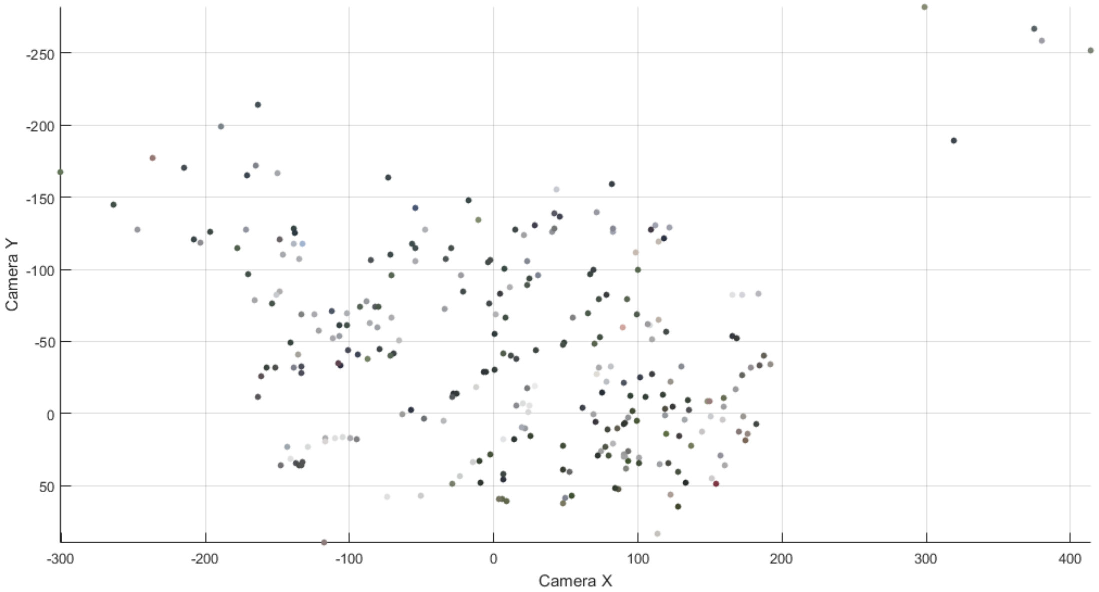
Example



- Two views with known relative pose
- Matching feature points
- After filtering by the epipolar constraint
  - Keeping matches that are within ± 0.5 pixels of the epipolar line

Example

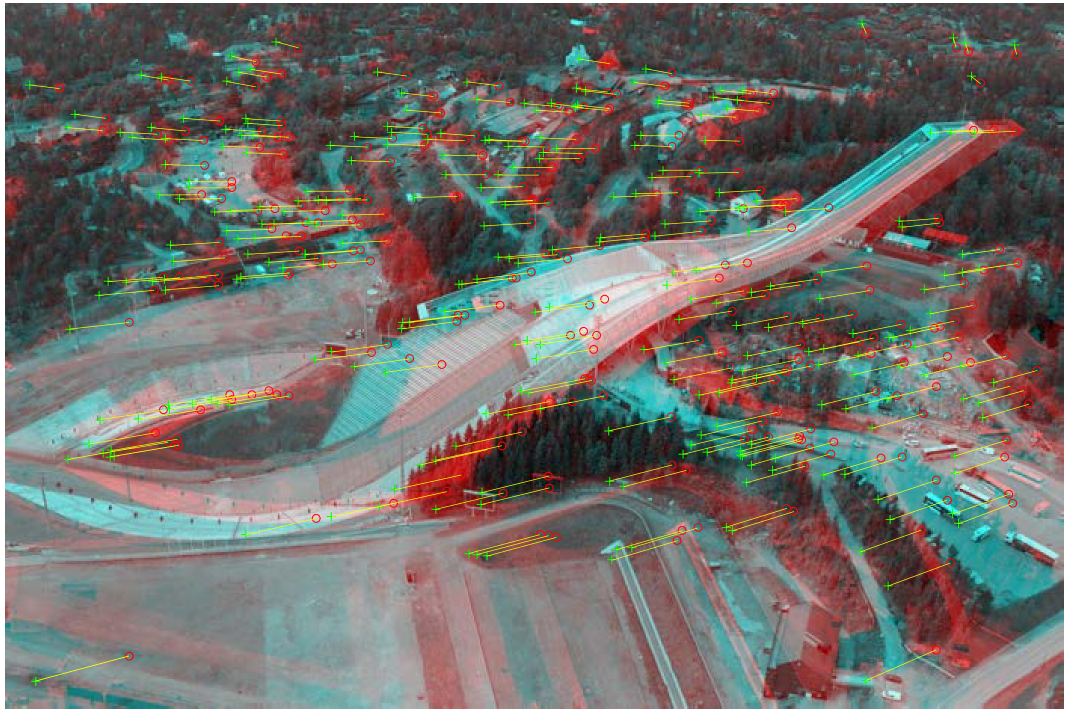


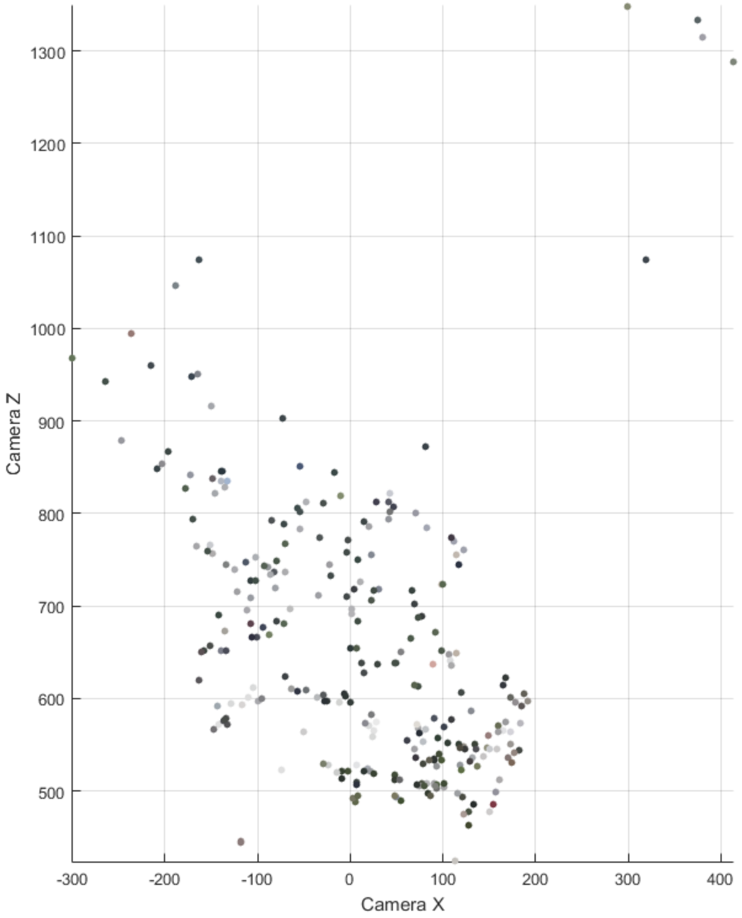


- Two views with known relative pose
- Matching feature points
- After filtering by the epipolar constraint
  - Keeping matches that are within ± 0.5 pixels of the epipolar line

- Sparse 3D reconstruction of the scene by triangulation

Example

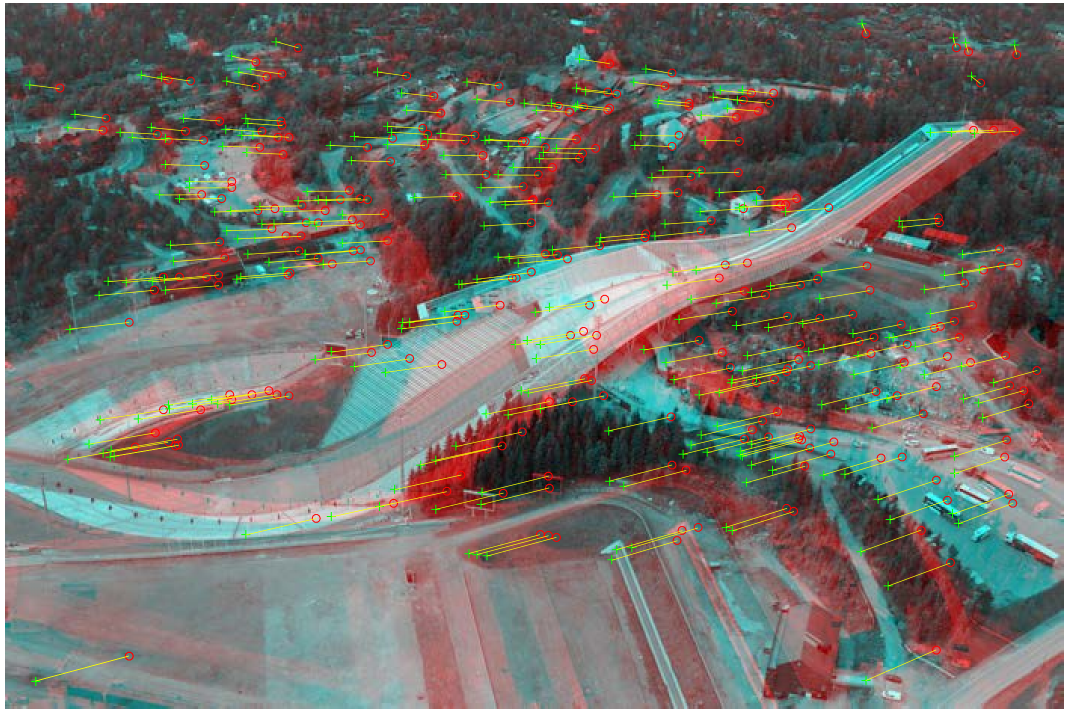


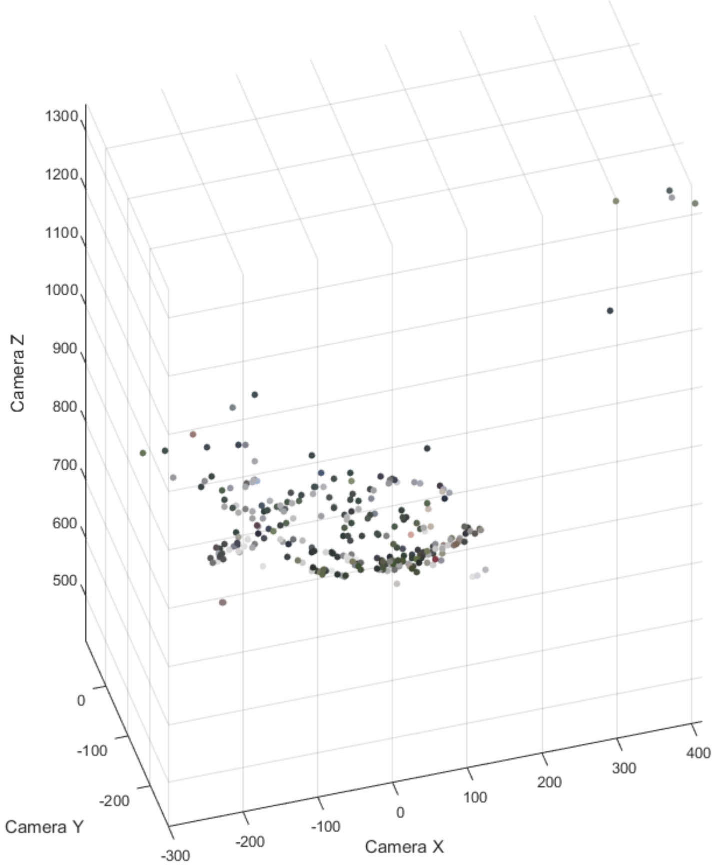


- Two views with known relative pose
- Matching feature points
- After filtering by the epipolar constraint
  - Keeping matches that are within ± 0.5 pixels of the epipolar line

- Sparse 3D reconstruction of the scene by triangulation

Example





- Two views with known relative pose
- Matching feature points
- After filtering by the epipolar constraint
  - Keeping matches that are within ± 0.5 pixels of the epipolar line

- Sparse 3D reconstruction of the scene by triangulation

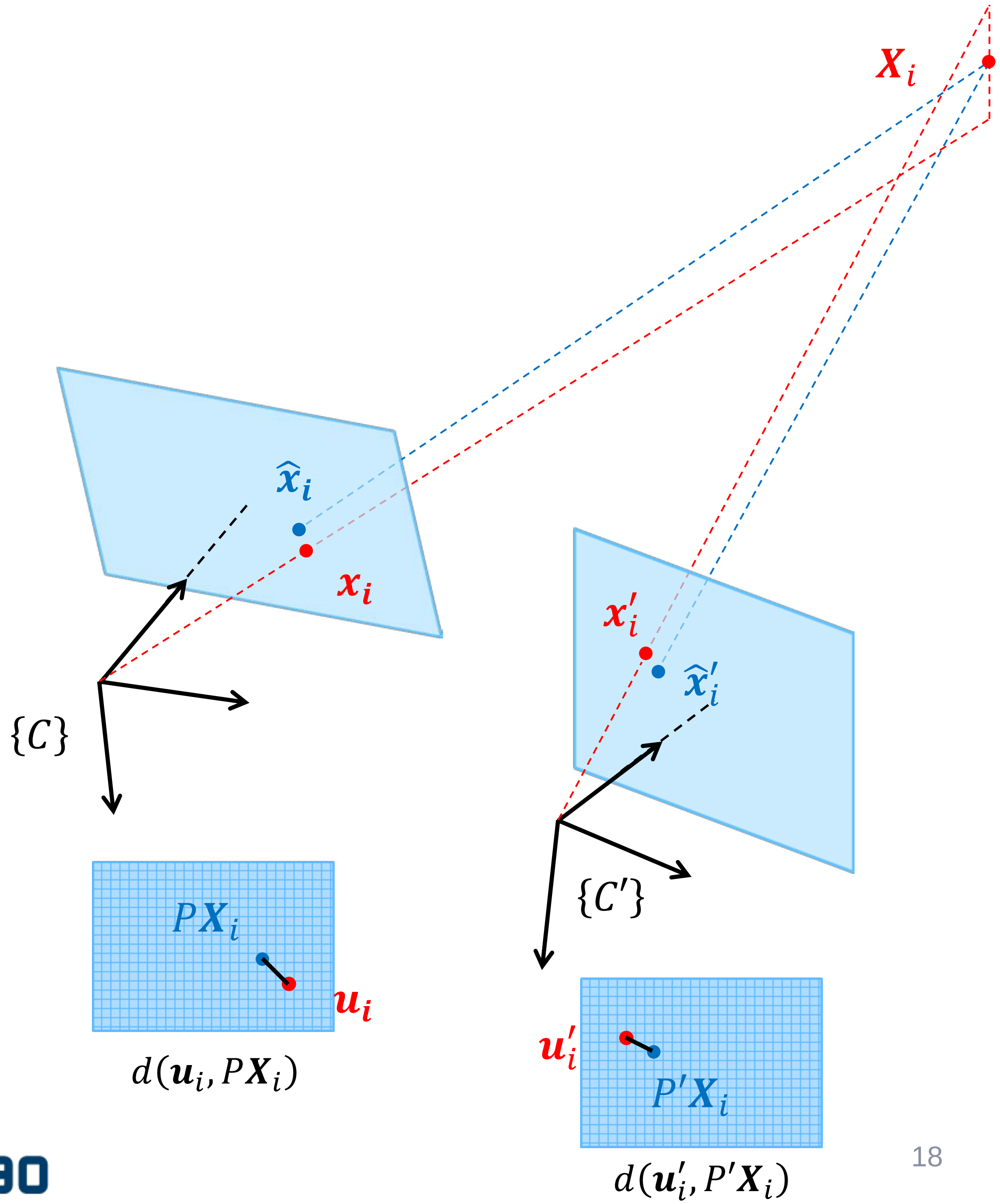
Non-linear triangulation
Minimizing a geometric error

- Compared to the previous algorithms it would be better to find the 3d point $\boldsymbol{X}_i$ that minimize a meaningful geometric error, like the reprojection error

$$\epsilon_i = d(\boldsymbol{u}_i, P\boldsymbol{X}_i)^2 + d(\boldsymbol{u}_i', P'\boldsymbol{X}_i)^2$$

- It can be shown that if the measurement noise in image points is Gaussian with mean equal to zero, the minimizing the reprojection error gives the Maximum Likelihood estimate of $\boldsymbol{X}_i$

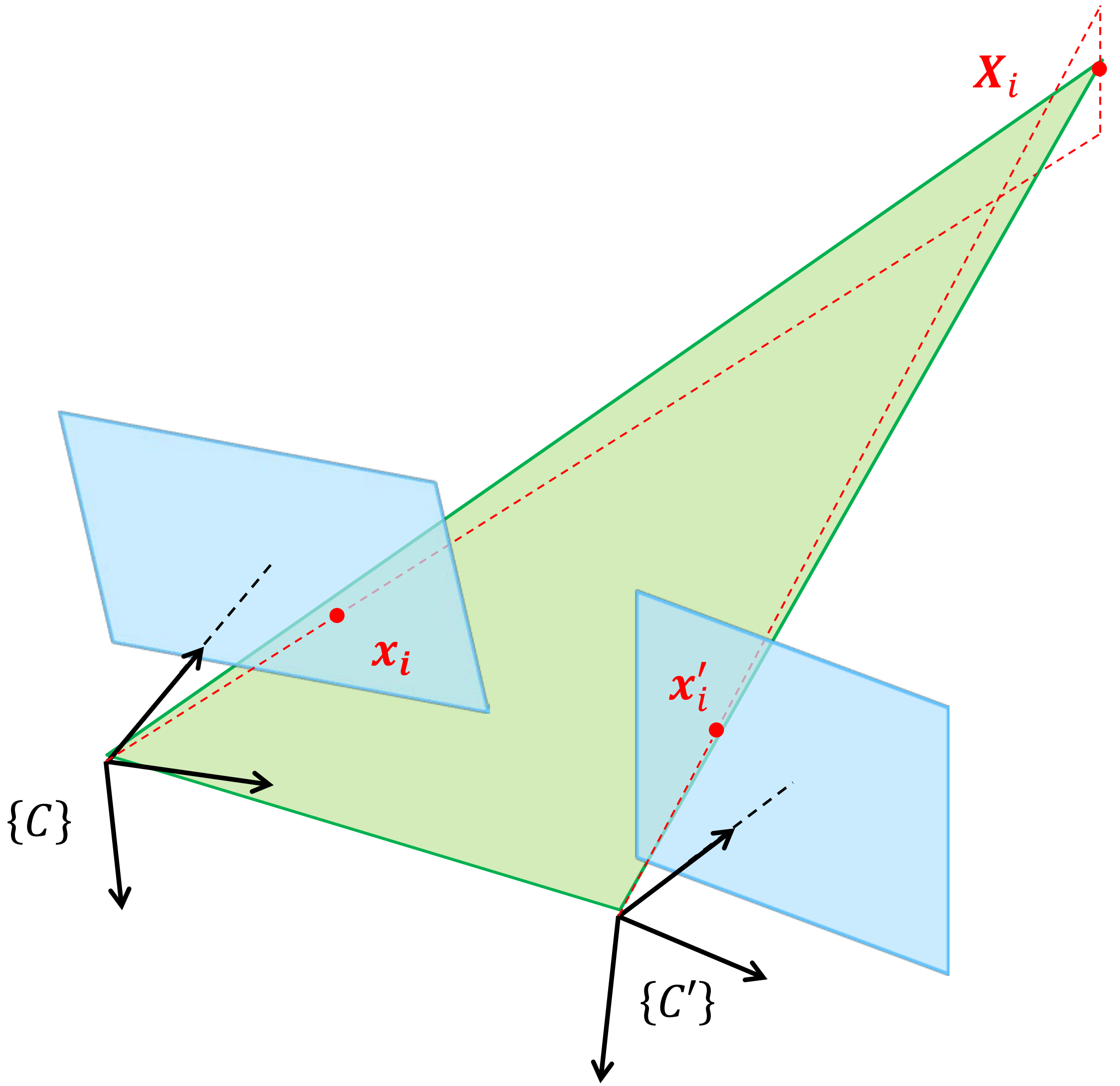
- At first glance, this minimization appears to be over the three parameters in $\boldsymbol{X}_i$, but under the assumption that P and P' are error free the problem can be reduced to a minimization over one parameter



Non-linear triangulation
Minimizing a geometric error

- If P and P' are error free, then the epipolar geometry is error free
  - We have a unique baseline which define all possible epipolar planes as a 1-parameter family
  - We have unique epipoles that all epipolar lines must pass through, so we have 1-parameter families of epipolar lines as well

- By requiring that both reprojected points lie on the same epipolar plane, the minimization problem can be reformulated in terms of the 1-parameter families of epipolar lines



Non-linear triangulation

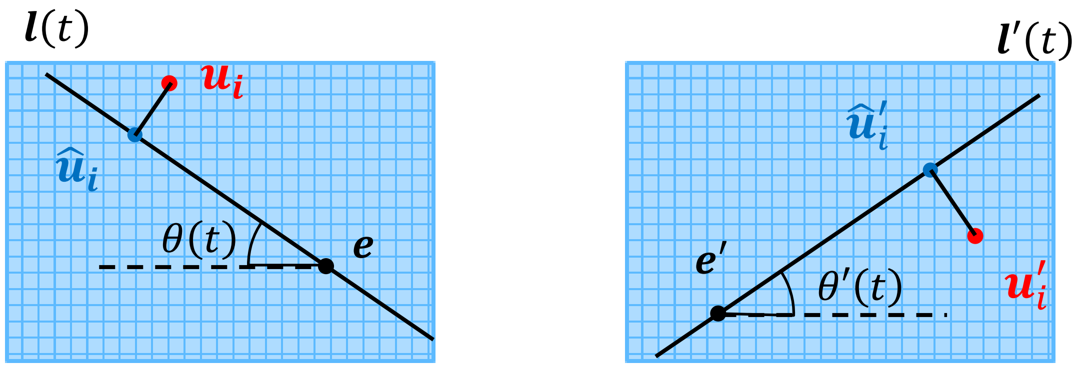
Minimizing a geometric error

- If P and P' are error free, then the epipolar geometry is error free
  - We have a unique baseline which define all possible epipolar planes as a 1-parameter family
  - We have unique epipoles that all epipolar lines must pass through, so we have 1-parameter families of epipolar lines as well

- By requiring that both reprojected points lie on the same epipolar plane, the minimization problem can be reformulated in terms of the 1-parameter families of epipolar lines

$$\epsilon_i = d(\boldsymbol{u}_i, \boldsymbol{l}(t))^2 + d\left(\boldsymbol{u}_i', \boldsymbol{l}'(t)\right)^2$$

- To find the t that minimize the reprojection error one has to minimize a 6th degree polynomial in t



- More details about this method and comparison with other methods can be found in the 1997 paper *Triangulation* by R. I. Hartley and P. Sturm

Summary

- **Triangulation** – Estimate a 3D point $\boldsymbol{X}_i$ for a noisy 2D correspondence under the assumption that camera matrices P and P' are known

- **Minimal 3D error** – Choose $\boldsymbol{X}_i$ to be the mid-point between back projected image points

- **Minimal algebraic error** – Combine the two perspective models to get a homogeneous system of linear equations, then determine $\boldsymbol{X}_i$ by SVD

- **Minimal reprojection error** – Determine the epipolar plane (and points $\hat{\boldsymbol{u}}_i$ and $\hat{\boldsymbol{u}}_i'$) that minimize the reprojection error by minimizing a 6^{th} order polynomial

- Additional reading
  - Szeliski: 7.1

- Optional reading
  - R. I. Hartley and P. Sturm, *Triangulation* (1997)