UiO : Department of Technology Systems
University of Oslo

Lecture 10.2
Building a consistent map from observations

Trym Vegard Haavardsholm

Part I

RECAP ON NONLINEAR LEAST-SQUARES (WITH UPDATED NOTATION)

Linear least squares

When the equations $e(\mathbf{x})$ are linear,
we can obtain an objective function on the form

$$f(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2 = \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|^2$$

A solution is required to have zero gradient:

$$\nabla f(\mathbf{x}^*) = 2\mathbf{A}^T \left(\mathbf{A}\mathbf{x}^* - \mathbf{b} \right) = \mathbf{0}$$

This results in the **normal equations**,

$$\mathbf{A}^T \mathbf{A}\mathbf{x}^* = \mathbf{A}^T \mathbf{b}$$

$$\mathbf{x}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

which can be solved with Cholesky- or QR factorization.

Linear least squares

When the equations $e(\mathbf{x})$ are linear,
we can obtain an objective function on the form

$$f(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2 = \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|^2$$

A solution is required to have zero gradient:

$$\nabla f(\mathbf{x}^*) = 2\mathbf{A}^T \left(\mathbf{A}\mathbf{x}^* - \mathbf{b} \right) = \mathbf{0}$$

This results in the **normal equations**,

$$\mathbf{A}^T \mathbf{A}\mathbf{x}^* = \mathbf{A}^T \mathbf{b}$$
$$\mathbf{x}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

which can be solved with Cholesky- or QR factorization.

Matlab example:

```
x = A\b;
```

Eigen example:

```
A.colPivHouseholderQr().solve(b);
```

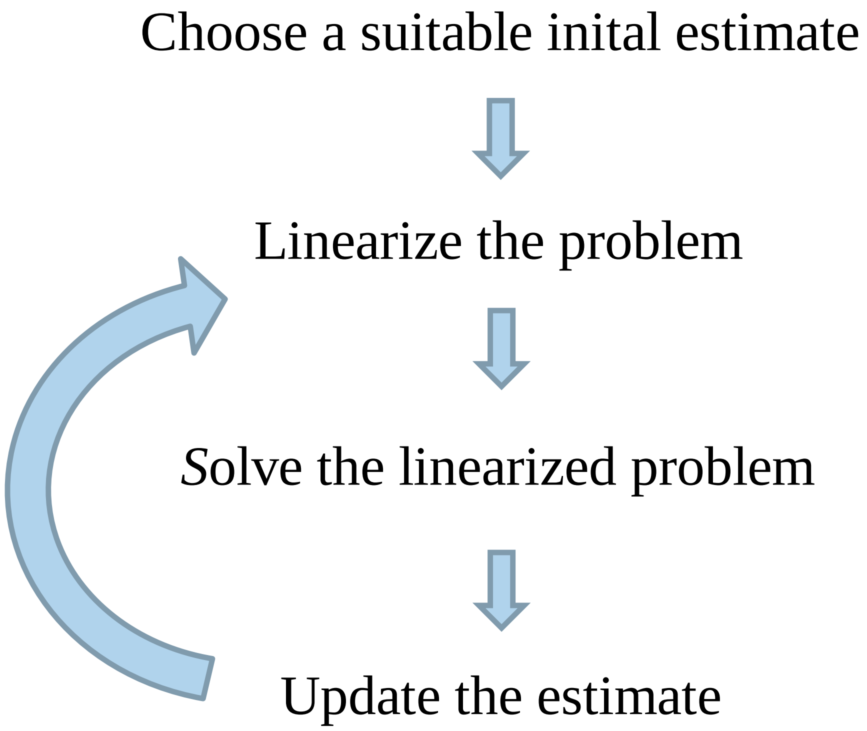
Read more about LLS:

- http://vmls-book.stanford.edu/vmls.pdf

Nonlinear least squares

When the equations $e(\mathbf{x})$ are nonlinear,
we have a **nonlinear least squares** problem.

They cannot be solved directly,
but require an iterative procedure
starting from a suitable initial estimate.



Nonlinear MAP inference for state estimation

We will use nonlinear least squares to solve **state estimation problems** based on **measurements** and corresponding **measurement models**

Let X be the set of all unknown state variables,
and Z be the set of all measurements.

We are interested in estimating the unknown state variables X, given the measurements Z.
The **Maximum a Posteriori estimate** is given by:

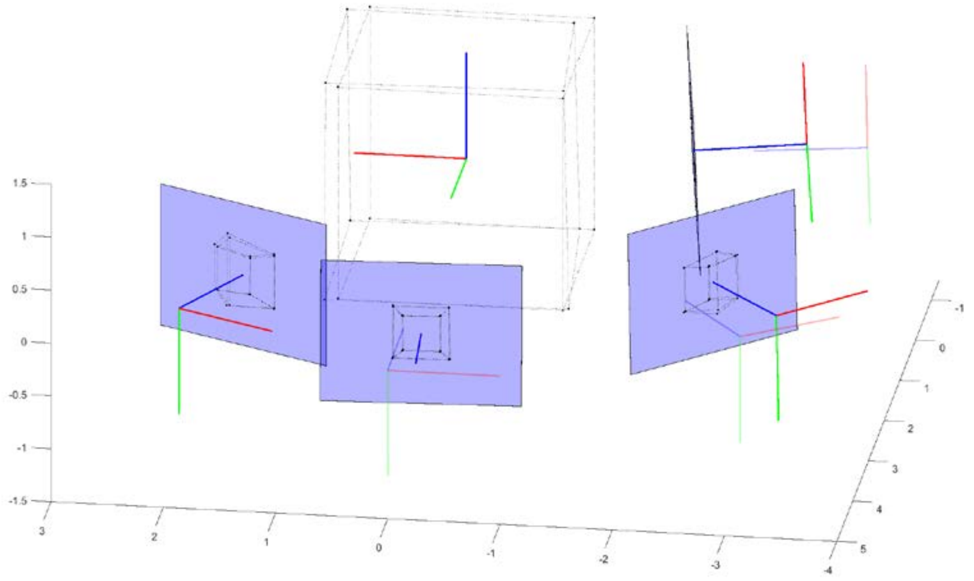
$$X^{MAP} = \underset{X}{\operatorname{argmax}}\, p(X \mid Z)$$

State variables

A **state variable** $\mathbf{x}$ is typically used to describe the physical state of an object.

We can estimate several state variables at once
by concatenating all the variables into the vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_p \end{bmatrix}$$



The equations $e_i(\mathbf{x})$ can be defined to operate
on one or more of these p state variables.

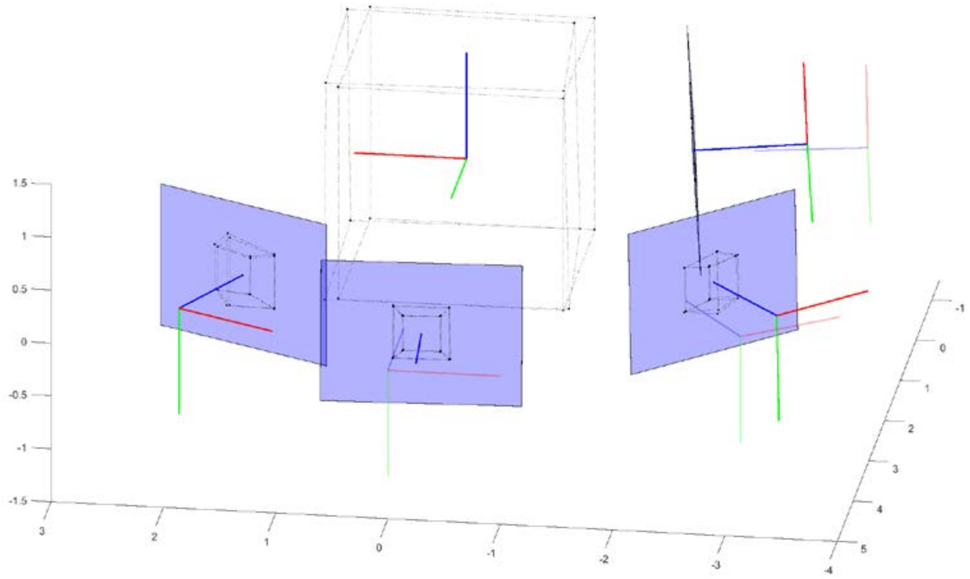
State variables

A **state variable** $\mathbf{x}$ is typically used to describe the physical state of an object.

We can estimate several state variables at once by concatenating all the variables into the vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_p \end{bmatrix}$$

The equations $e_i(\mathbf{x})$ can be defined to operate on one or more of these p state variables.



How can we represent both points and poses as states?

Orientations and poses lie on manifolds

Orientations and poses lie on manifolds
in higher-dimensional spaces

This makes it complicated to add increments,
represent uncertainty and perform differentiation

Example:

$$\mathbf{R} \in SO(3)$$

$$\delta\mathbf{R} \in \mathbb{R}^{3\times3}$$

$$\mathbf{R} + \delta\mathbf{R} \notin SO(3)$$

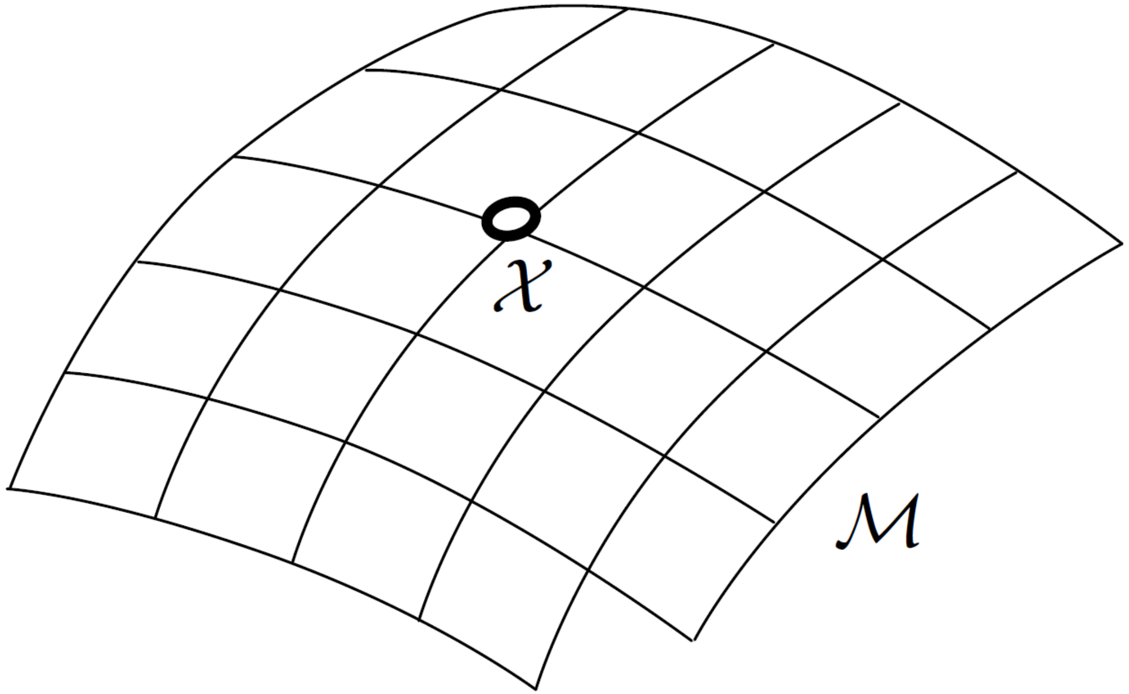


Image source: Solà, J., Deray, J., & Atchuthan, D. (n.d.). A micro Lie theory for state estimation in robotics
(Cropped and edited; licensed under CC BY-NC-SA 4.0)

Lie theory lets us work on these manifolds

Orientations and poses are **matrix Lie groups**

A Lie group is a **group** on a **smooth manifold**

Lie theory describes the **tangent space** around elements of a Lie group,
and defines **exact mappings** between
the tangent space and the manifold

The tangent space is a **vector space** with the same dimension as the number of degrees of freedom of the group transformations

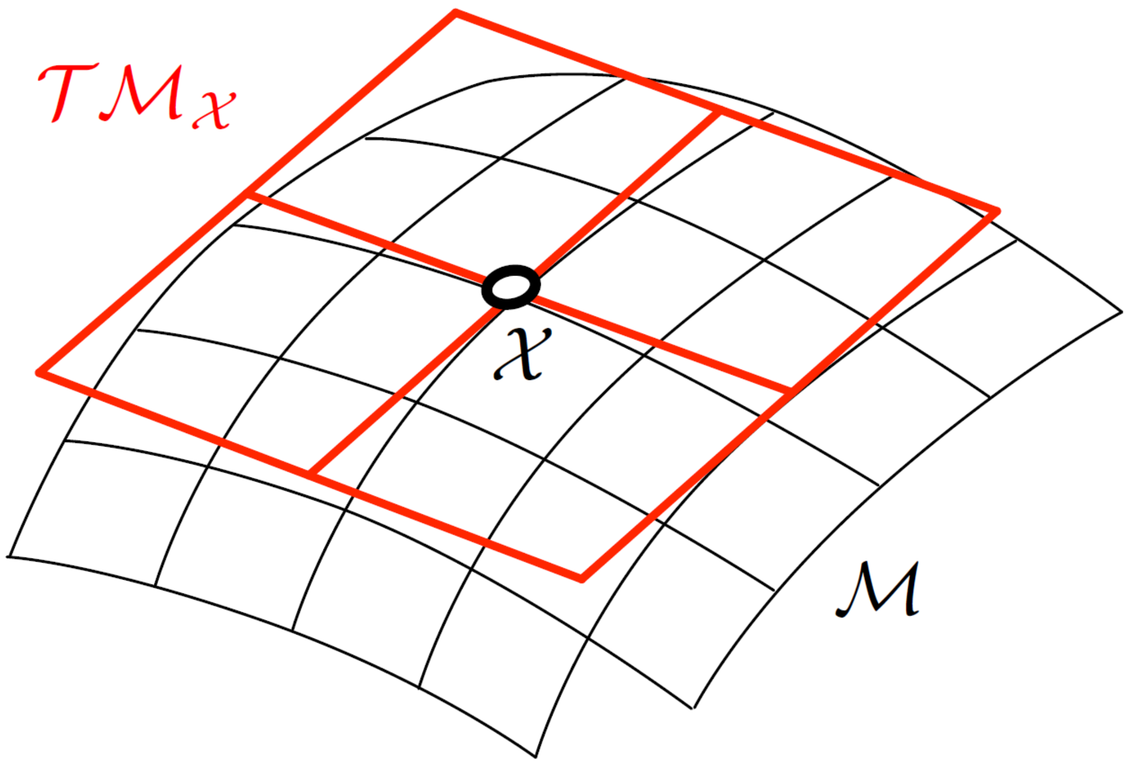


Image source: Solà, J., Deray, J., & Atchuthan, D. (n.d.). A micro Lie theory for state estimation in robotics
(Cropped; licensed under CC BY-NC-SA 4.0)

The exponential map

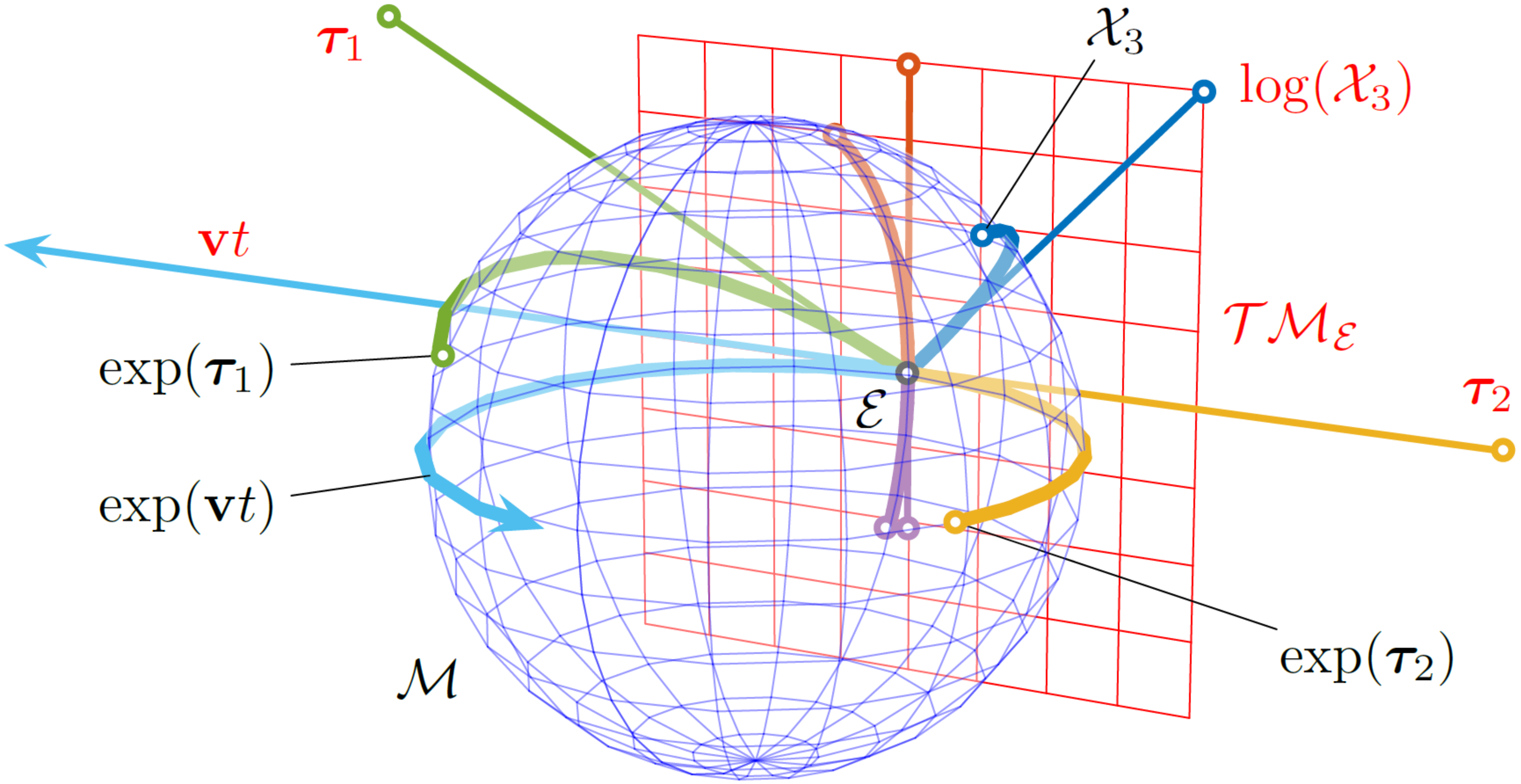


Image source: Solà, J., Deray, J., & Atchuthan, D. (n.d.). A micro Lie theory for state estimation in robotics (licensed under CC BY-NC-SA 4.0)

Plus and minus operators

It is convenient to express perturbations using plus and minus operators.

The **right plus and minus operators** are defined as:

$$\mathcal{Y} = \mathcal{X} \oplus {}^{\mathcal{X}}\boldsymbol{\tau} \triangleq \mathcal{X} \circ \operatorname{Exp}({}^{\mathcal{X}}\boldsymbol{\tau}) \in \mathcal{M}$$
$${}^{\mathcal{X}}\boldsymbol{\tau} = \mathcal{Y} \ominus \mathcal{X} \triangleq \operatorname{Log}(\mathcal{X}^{-1} \circ \mathcal{Y}) \in \mathcal{TM}_{\mathcal{X}}$$

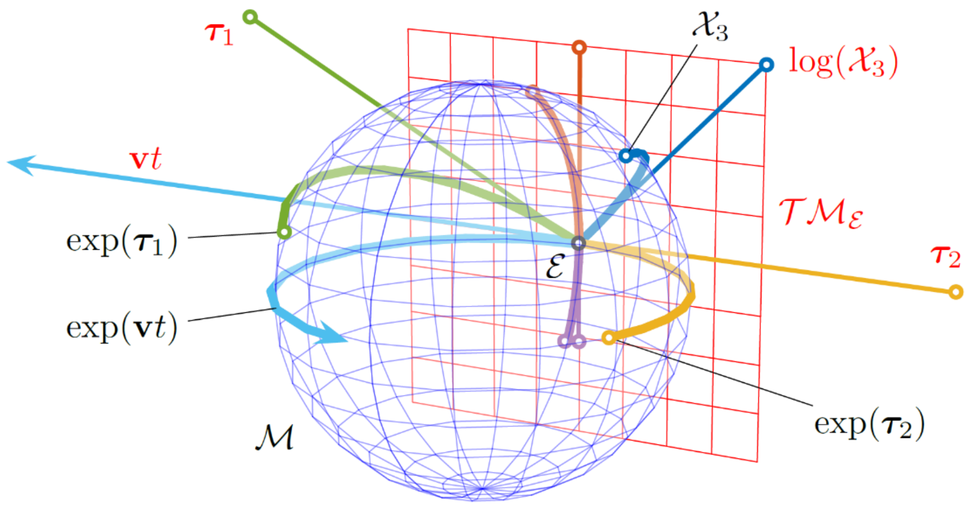


Image source: Solà, J., Deray, J., & Atchuthan, D. (n.d.). A micro Lie theory for state estimation in robotics (licensed under CC BY-NC-SA 4.0)

Concatenated set of state variables

Concatenation of state variables over a composite manifold
and the corresponding concatenation of tangent space vectors

$$\underline{\mathcal{X}} \triangleq \begin{Bmatrix} \mathcal{X}_1 \\ \vdots \\ \mathcal{X}_p \end{Bmatrix} \in \mathcal{M} \qquad \underline{\boldsymbol{\tau}} \triangleq \begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_p \end{bmatrix} \in \mathbb{R}^m \qquad \begin{aligned} &\mathcal{X}_i \in \mathcal{M}_i \\ &\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_p\} \\ &\boldsymbol{\tau}_i \in T\mathcal{M}_i \end{aligned}$$

Plus and minus for the concatenated state variable

$$\underline{\mathcal{X}} \oplus \underline{\boldsymbol{\tau}} \triangleq \begin{Bmatrix} \mathcal{X}_1 \oplus \boldsymbol{\tau}_1 \\ \vdots \\ \mathcal{X}_p \oplus \boldsymbol{\tau}_p \end{Bmatrix} \in \mathcal{M} \qquad \underline{\mathcal{Y}} \ominus \underline{\mathcal{X}} \triangleq \begin{bmatrix} \mathcal{Y}_1 \ominus \mathcal{X}_1 \\ \vdots \\ \mathcal{Y}_p \ominus \mathcal{X}_p \end{bmatrix} \in \mathbb{R}^m$$

Concatenated set of state variables

We define $\underline{X}_i$ to be the concatenated set of state variables taken as input by the *i*-th equation $e_i(\underline{X}_i)$.

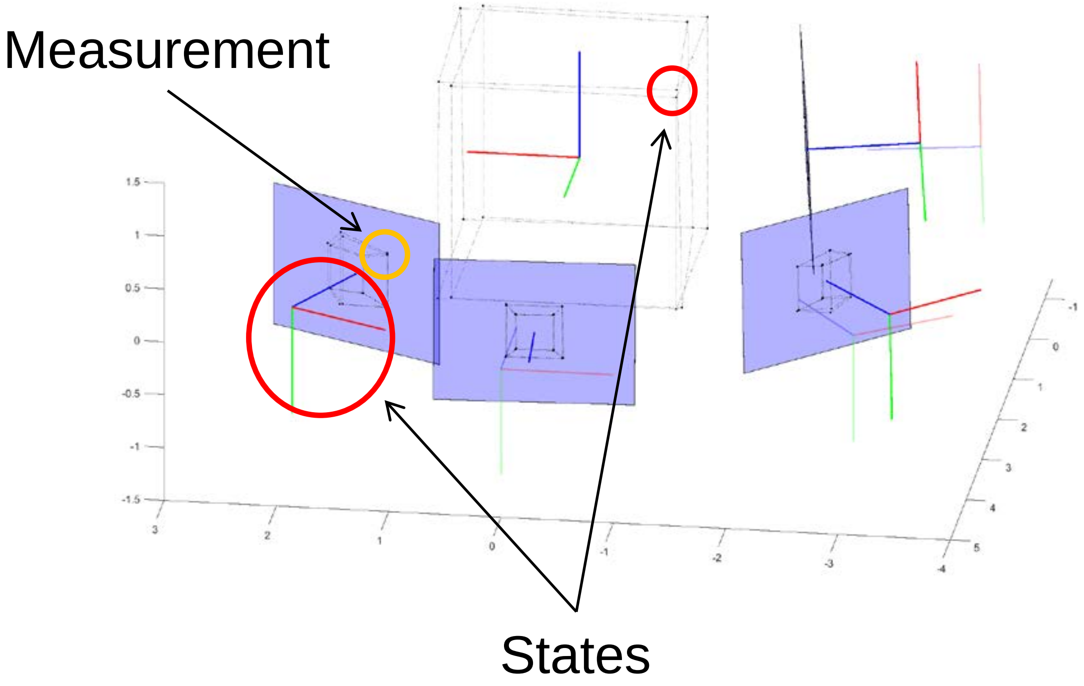
Concatenated set of state variables

We define $\underline{X}_i$ to be the concatenated set of state variables taken as input by the *i*-th equation $e_i(\underline{X}_i)$.

Example:

$$e_{ij}(\underline{X}_{ij}) = e_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w) = \pi(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{u}_j^i$$

$$\underline{X}_{ij} = \begin{Bmatrix} \mathbf{T}_{wc_i} \\ \mathbf{x}_j^w \end{Bmatrix}$$



Concatenated set of state variables

We define $\underline{\mathrm{X}}_i$ to be the concatenated set of state variables taken as input by the *i*-th equation $e_i(\underline{\mathrm{X}}_i)$.

We can then define the objective function over all state variables

$$f(\underline{\mathrm{X}}) = \left\| e(\underline{\mathrm{X}}) \right\|^2 = \sum_{i=1}^{n} \left\| e_i(\underline{\mathrm{X}}_i) \right\|^2$$

Nonlinear MAP inference for state estimation

Measurement model:

$$\mathbf{z}_i = h_i(\underline{\mathrm{X}}_i) + \eta_i, \qquad \eta_i \sim N(\mathbf{0}, \mathbf{\Sigma}_i)$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(\underline{\mathrm{X}}_i)$$

Measurement error function:

$$e_i(\underline{\mathrm{X}}_i) = h_i(\underline{\mathrm{X}}_i) - \mathbf{z}_i$$

Objective function:

$$f(\underline{\mathrm{X}}) = \sum_{i=1}^{n} \left\| h_i(\underline{\mathrm{X}}_i) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2$$

where $\left\| \mathbf{e} \right\|_{\mathbf{\Sigma}}^2 = \mathbf{e}^T \mathbf{\Sigma}^{-1} \mathbf{e}$ is the squared Mahalanobis norm

Nonlinear MAP inference for state estimation

Measurement model:

$$\mathbf{z}_i = h_i(\underline{\mathrm{X}}_i) + \eta_i, \qquad \eta_i \sim N(\mathbf{0}, \mathbf{\Sigma}_i)$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(\underline{\mathrm{X}}_i)$$

Measurement error function:

$$e_i(\underline{\mathrm{X}}_i) = h_i(\underline{\mathrm{X}}_i) - \mathbf{z}_i$$

Objective function:

$$f(\underline{\mathrm{X}}) = \sum_{i=1}^{n} \left\| h_i(\underline{\mathrm{X}}_i) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2$$

This results in the nonlinear least squares problem:

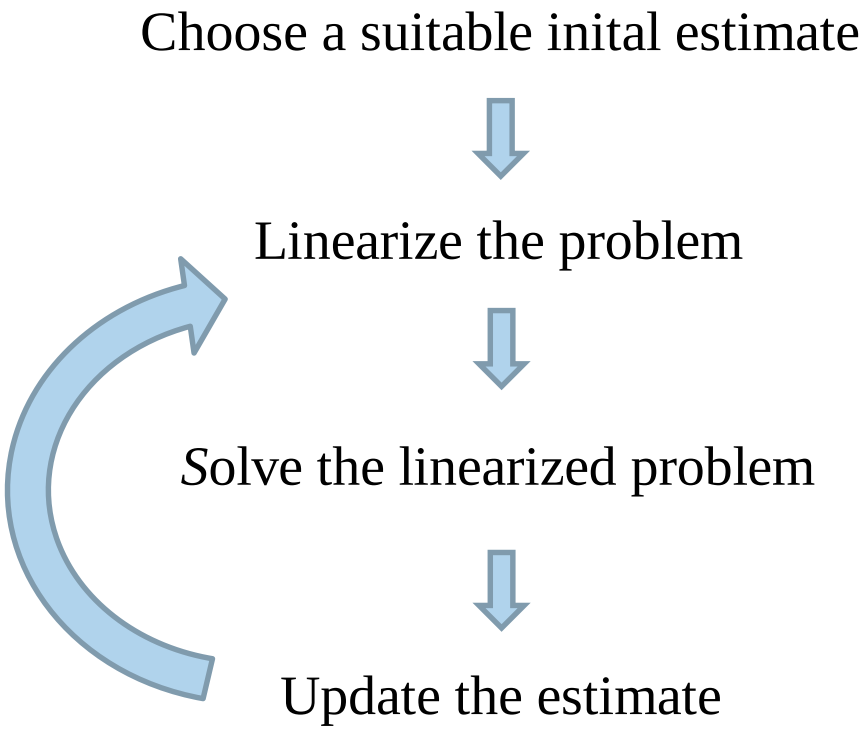
$$\underline{\mathrm{X}}^* = \underset{\underline{\mathrm{X}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| h_i(\underline{\mathrm{X}}_i) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2$$

It turns out that the nonlinear least squares solution to this problem is the MAP estimate!

Nonlinear least squares

When the equations $e_i(\underline{\mathrm{X}}_i) = h_i(\underline{\mathrm{X}}_i) - \mathbf{z}_i$ are nonlinear,
we have a **nonlinear least squares** problem.

They cannot be solved directly,
but require an iterative procedure
starting from a suitable initial estimate.



Linearizing the problem

We can linearize the measurement prediction functions using **first order Taylor expansions** at the current estimates $\underline{\hat{\mathcal{X}}}_i$:

$$h_i(\underline{\mathcal{X}}_i) = h_i(\underline{\hat{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \approx h_i(\underline{\hat{\mathcal{X}}}_i) + \mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i$$

where the **measurement Jacobian** $\mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i}$ is

$$\mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i} \triangleq \left. \frac{\partial h_i(\underline{\mathcal{X}}_i)}{\partial \underline{\mathcal{X}}_i} \right|_{\underline{\hat{\mathcal{X}}}_i}$$

and

$$\underline{\boldsymbol{\tau}}_i \triangleq \underline{\mathcal{X}}_i \ominus \underline{\hat{\mathcal{X}}}_i$$

is the **state update vector**.

Linearizing the problem

This leads to the linearized measurement error function

$$e_i(\underline{\mathcal{X}}_i) = e_i(\hat{\underline{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \approx h_i(\hat{\underline{\mathcal{X}}}_i) + \mathbf{J}^{h_i}_{\hat{\underline{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i - \mathbf{z}_i$$

Linearizing the problem

The linearized objective function is then given by

$$
\begin{aligned}
f(\underline{\mathcal{X}}) = f(\hat{\underline{\mathcal{X}}} \oplus \underline{\boldsymbol{\tau}}) &= \sum_{i=1}^{n} \left\| e_i(\hat{\underline{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&\approx \sum_{i=1}^{n} \left\| h_i(\hat{\underline{\mathcal{X}}}_i) + \mathbf{J}_{\hat{\underline{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{J}_{\hat{\underline{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \left(\mathbf{z}_i - h_i(\hat{\underline{\mathcal{X}}}_i) \right) \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&= \sum_{i=1}^{n} \left\| \boldsymbol{\Sigma}_i^{-1/2} \mathbf{J}_{\hat{\underline{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \boldsymbol{\Sigma}_i^{-1/2} \left(\mathbf{z}_i - h_i(\hat{\underline{\mathcal{X}}}_i) \right) \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{A}_i \underline{\boldsymbol{\tau}}_i - \mathbf{b}_i \right\|^2 \\
&= \left\| \mathbf{A} \underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2
\end{aligned}
$$

Solving the linearized problem

The linearized objective function is then given by

$$
\begin{aligned}
f(\underline{\mathcal{X}}) = f(\hat{\underline{\mathcal{X}}} \oplus \underline{\boldsymbol{\tau}}) &= \sum_{i=1}^{n} \left\| e_i(\hat{\underline{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&\approx \sum_{i=1}^{n} \left\| h_i(\hat{\underline{\mathcal{X}}}_i) + \mathbf{J}_{\hat{\underline{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \mathbf{z}_i \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{J}_{\hat{\underline{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \left(\mathbf{z}_i - h_i(\hat{\underline{\mathcal{X}}}_i) \right) \right\|_{\boldsymbol{\Sigma}_i}^2 \\
&= \sum_{i=1}^{n} \left\| \boldsymbol{\Sigma}_i^{-1/2} \mathbf{J}_{\hat{\underline{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \boldsymbol{\Sigma}_i^{-1/2} \left(\mathbf{z}_i - h_i(\hat{\underline{\mathcal{X}}}_i) \right) \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{A}_i \underline{\boldsymbol{\tau}}_i - \mathbf{b}_i \right\|^2 \\
&= \left\| \mathbf{A} \underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2
\end{aligned}
$$

We can solve the linearized problem as a linear least squares problem using the normal equations

$$
\mathbf{A}^T \mathbf{A} \underline{\boldsymbol{\tau}}^* = \mathbf{A}^T \mathbf{b}
$$

Solving the nonlinear problem

We solve the nonlinear least-squares problem
by iteratively solving the linearized system:

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

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

$$\underline{\boldsymbol{\tau}}^{*} \leftarrow \mathit{S}\text{olve } \underset{\underline{\boldsymbol{\tau}}}{\operatorname{argmin}} \left\| \mathbf{A}\underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^{2}$$

$$\hat{\underline{\mathrm{X}}}^{t+1} \leftarrow \hat{\underline{\mathrm{X}}}^{t} \oplus \underline{\boldsymbol{\tau}}^{*}$$

The Gauss-Newton algorithm

Data: An objective function $f(\underline{\mathcal{X}})$ and a good initial state estimate $\hat{\underline{\mathcal{X}}}^0$
Result: An estimate for the states $\hat{\underline{\mathcal{X}}}$

for $t = 0, 1, \dots, t^{max}$ **do**
    $\mathbf{A}, \mathbf{b} \leftarrow$ Linearise $f(\underline{\mathcal{X}})$ at $\hat{\underline{\mathcal{X}}}^t$
    $\underline{\boldsymbol{\tau}} \leftarrow$ Solve the linearised problem $\mathbf{A}^\top \mathbf{A} \underline{\boldsymbol{\tau}} = \mathbf{A}^\top \mathbf{b}$
    $\hat{\underline{\mathcal{X}}}^{t+1} \leftarrow \hat{\underline{\mathcal{X}}}^t \oplus \underline{\boldsymbol{\tau}}$

    if $f(\hat{\underline{\mathcal{X}}}^{t+1})$ *is very small or* $\hat{\underline{\mathcal{X}}}^{t+1} \approx \hat{\underline{\mathcal{X}}}^t$ **then**
        $\hat{\underline{\mathcal{X}}} \leftarrow \hat{\underline{\mathcal{X}}}^{t+1}$
        return
    end
end

Part II

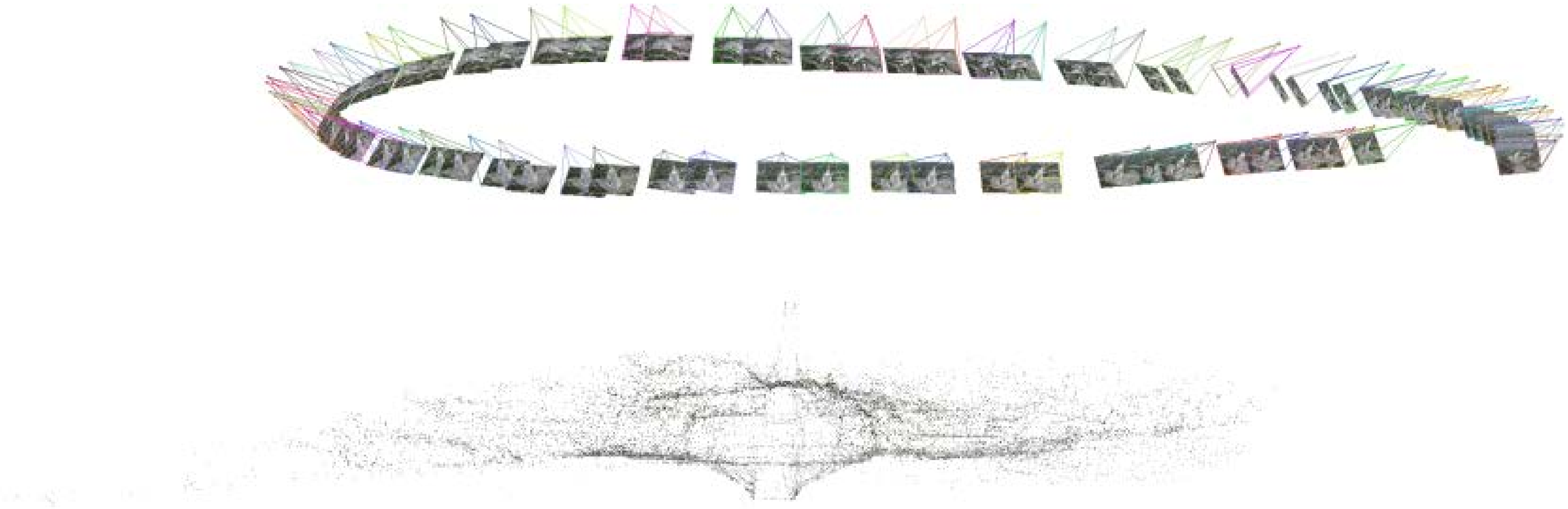
BUNDLE ADJUSTMENT

Bundle adjustment

Bundle Adjustment (BA)

Estimating the imaging geometry
based on minimizing reprojection error

- Motion-only BA
- Structure-only BA
- Full BA

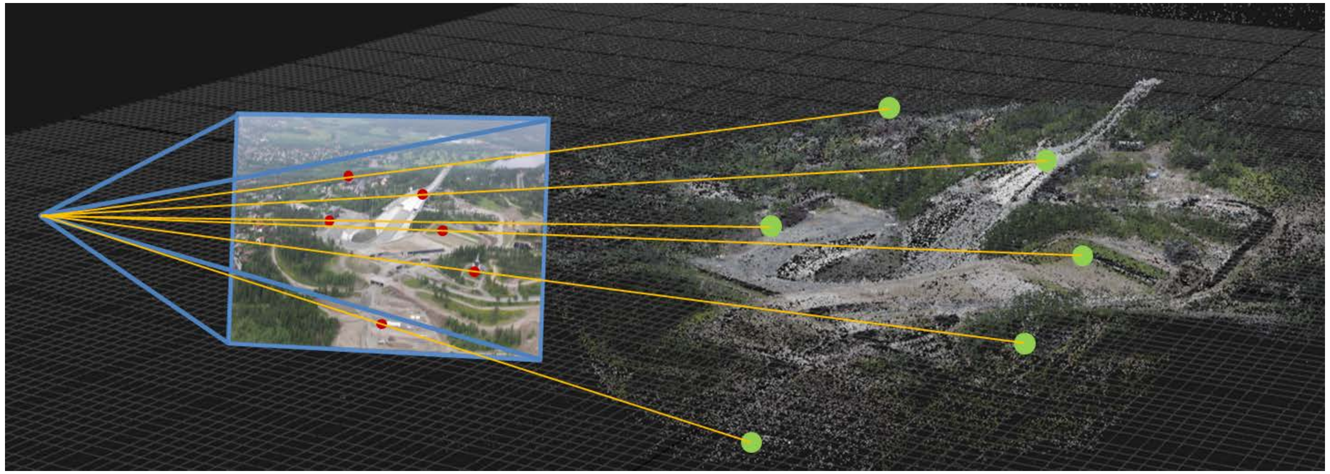


Pose estimation by minimizing reprojection error

Minimize **geometric error** over the **camera pose** given **known structure**

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

$$\mathbf{T}_{wc}^{*} = \underset{\mathbf{T}_{wc}}{\operatorname{argmin}} \sum_{j} \left\| \pi(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_{j}^{w}) - \mathbf{u}_{j} \right\|^{2}$$



Pose estimation by minimizing reprojection error

Given:

- World points $\mathbf{x}_j^w$

Measurements:

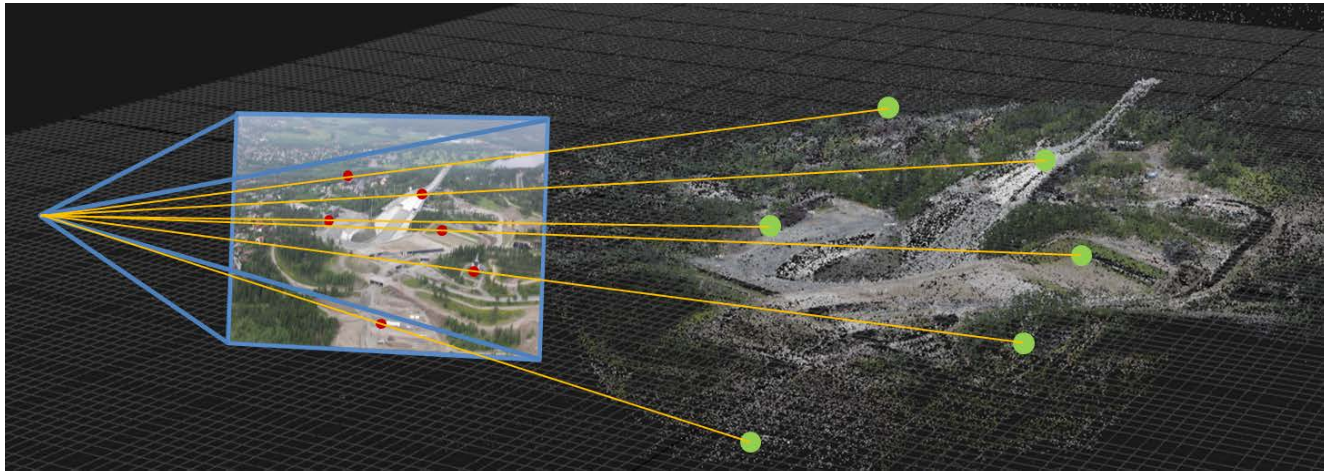
- Correspondences $\mathbf{u}_j \leftrightarrow \mathbf{x}_j^w$ with measurement noise $\mathbf{\Sigma}_j$

State we wish to estimate:

- Camera pose $\mathbf{T}_{wc}$

Initial estimate:

- PnP (P3P, EPnP, …)
- Motion model



Applying the MAP framework

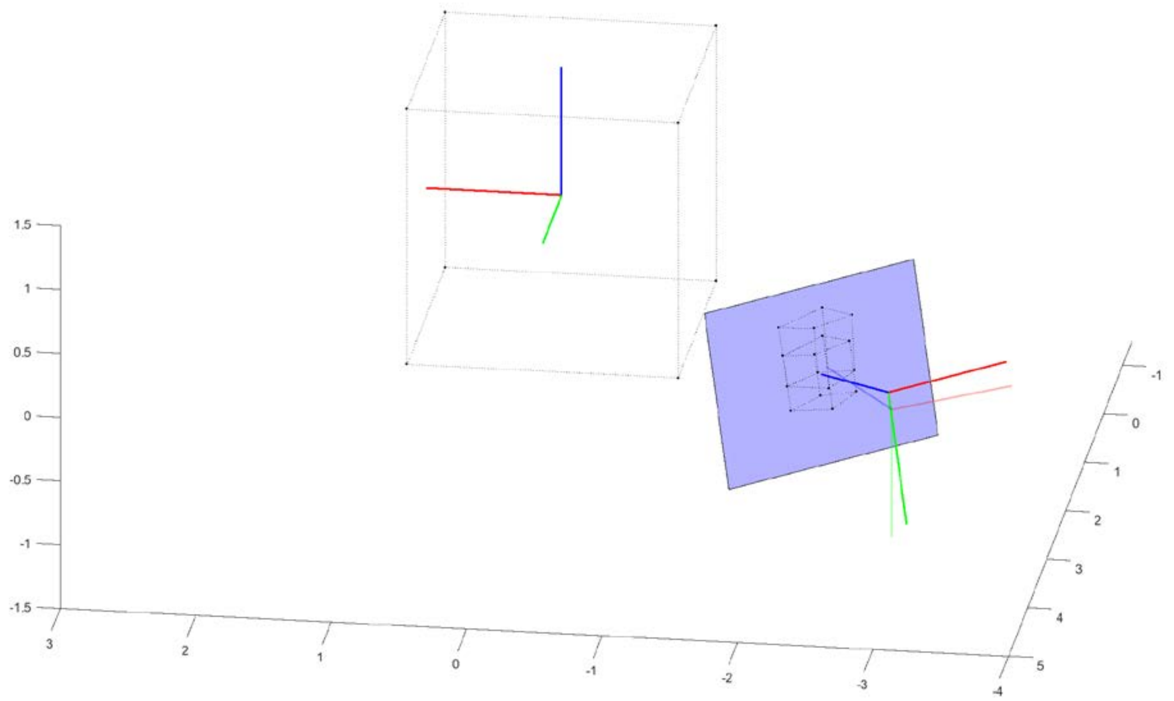
For simplicity,
we pre-calibrate to normalized image coordinates (and propagate the noise)

This gives us the measurement prediction function

$$h_j(\mathbf{T}_{wc}) = \pi_n(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_j^w)$$

and measurement error function

$$e_j(\mathbf{T}_{wc}) = \pi_n(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{x}_{n\,j}$$



Applying the MAP framework

The measurement Jacobian is given by

$$
\begin{aligned}
\mathbf{J}^{h}_{\mathbf{T}_{wc}} &= \mathbf{J}^{\pi_n(\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w)}_{\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w} \mathbf{J}^{\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w}_{\mathbf{T}_{wc}^{-1}} \mathbf{J}^{\mathbf{T}_{wc}^{-1}}_{\mathbf{T}_{wc}} \\
&= \mathbf{J}^{\pi_n(\mathbf{x}^c)}_{\mathbf{x}^c} \mathbf{J}^{\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w}_{\mathbf{T}_{wc}^{-1}} \mathbf{J}^{\mathbf{T}_{wc}^{-1}}_{\mathbf{T}_{wc}} \\
&= \frac{1}{z^c}\begin{bmatrix} 1 & 0 & -x^c/z^c \\ 0 & 1 & -y^c/z^c \end{bmatrix} \begin{bmatrix} \mathbf{R}_{wc}^{\top} & -\mathbf{R}_{wc}[\mathbf{x}^w]_\times \end{bmatrix} \cdot - \begin{bmatrix} \mathbf{R}_{wc} & [\mathbf{t}_{wc}^w]_\times \mathbf{R}_{wc} \\ \mathbf{0} & \mathbf{R}_{wc} \end{bmatrix} \\
&= d\begin{bmatrix} 1 & 0 & -x_n \\ 0 & 1 & -y_n \end{bmatrix} \begin{bmatrix} -\mathbf{I} & [\mathbf{x}^c]_\times \end{bmatrix} \\
&= \begin{bmatrix} -d & 0 & dx_n & x_n y_n & -1 - x_n^2 & y_n \\ 0 & -d & dy_n & 1 + y_n^2 & -x_n y_n & -x_n \end{bmatrix},
\end{aligned}
$$

Applying the MAP framework

This results in the linearized weighted least squares problem

$$
\begin{aligned}
\boldsymbol{\xi}^* &= \arg\min_{\boldsymbol{\xi}} \sum_{j=1}^{n} \left\| \mathbf{A}_j \boldsymbol{\xi} - \mathbf{b}_j \right\|^2 \\
&= \arg\min_{\boldsymbol{\xi}} \left\| \mathbf{A} \boldsymbol{\xi} - \mathbf{b} \right\|^2 ,
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{A}_j &= \boldsymbol{\Sigma}_{n\,j}^{-1/2} \mathbf{J}_{\mathbf{T}_{wc}}^{h_j} \\
\mathbf{b}_j &= \boldsymbol{\Sigma}_{n\,j}^{-1/2} (\mathbf{x}_{n\,j} - h_j(\mathbf{T}_{wc})),
\end{aligned}
\qquad
\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} .
$$

Applying the MAP framework

For an example with *three points*,
the measurement Jacobian $\mathbf{A}$ and the prediction error $\mathbf{b}$ are

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}$$

Applying the MAP framework

The solution can be found by solving the normal equations

$$\left(\mathbf{A}^T\mathbf{A}\right)\boldsymbol{\xi}^* = \mathbf{A}^T\mathbf{b}$$

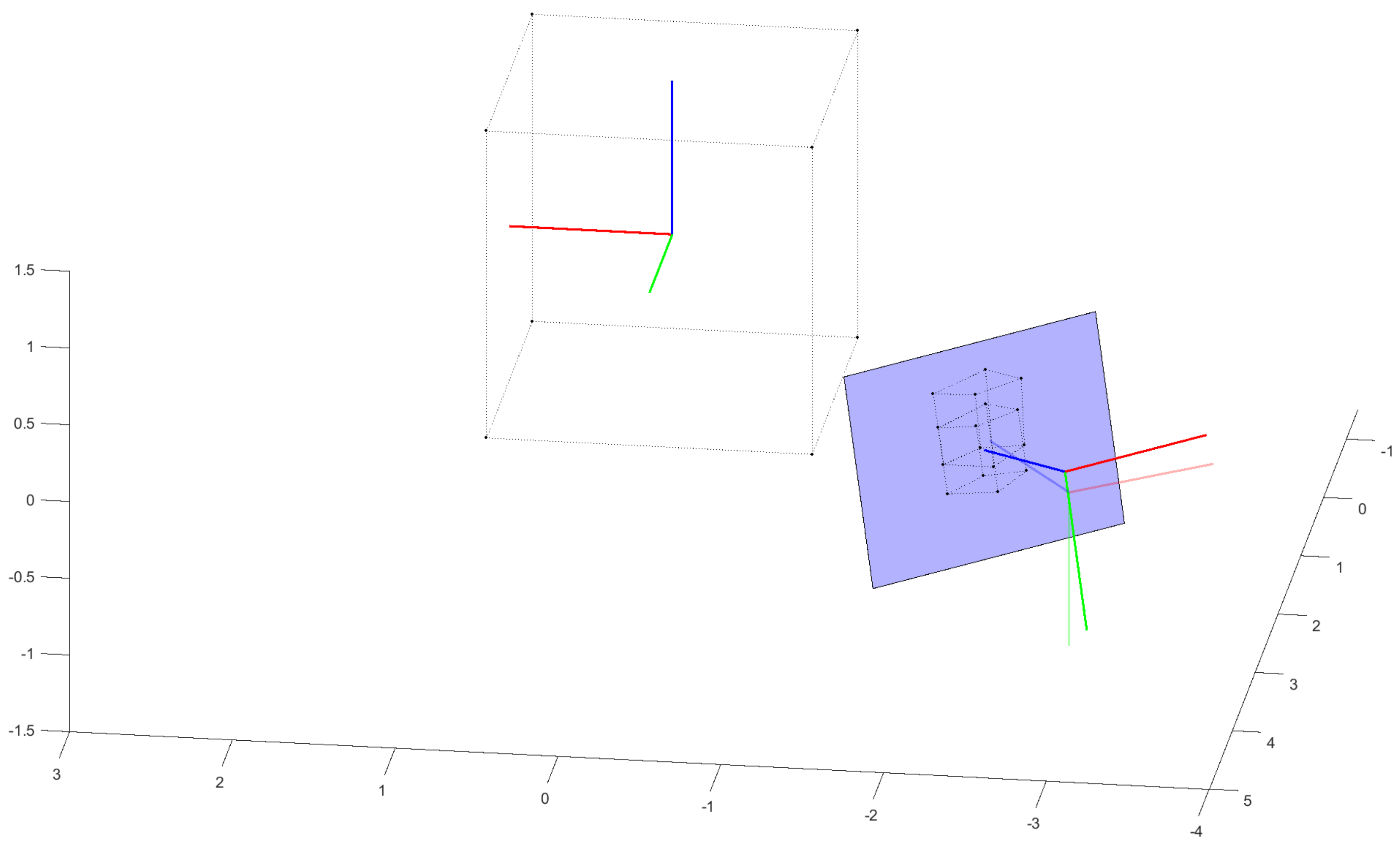
Choose a suitable inital estimate $\underline{\hat{X}}^{0}$

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

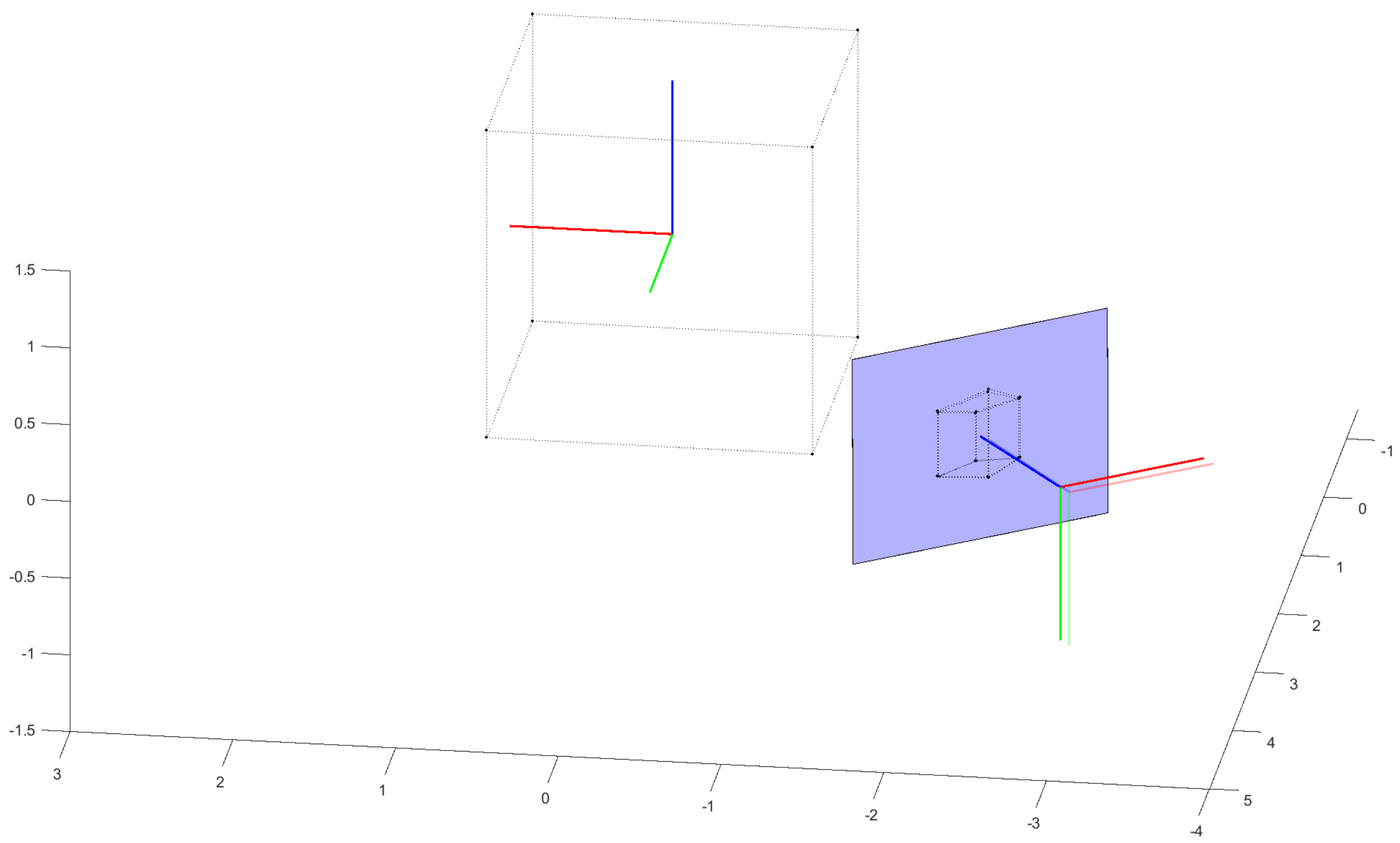
$$\underline{\boldsymbol{\tau}}^* \leftarrow \textit{Solve } \underset{\underline{\boldsymbol{\tau}}}{\operatorname{argmin}} \left\| \mathbf{A}\underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2$$

$$\underline{\hat{X}}^{t+1} \leftarrow \underline{\hat{X}}^{t} \oplus \underline{\boldsymbol{\tau}}^*$$

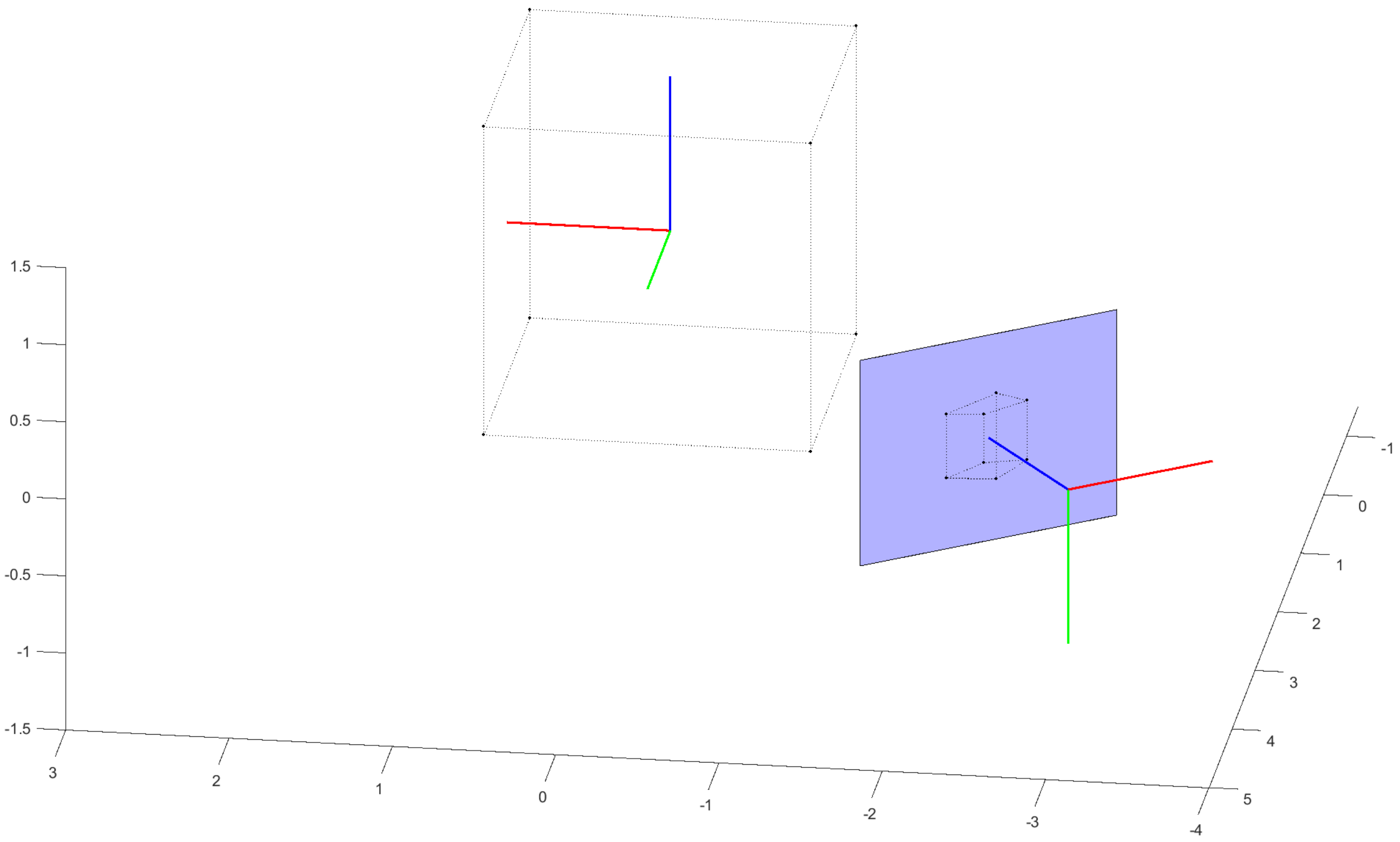
Example



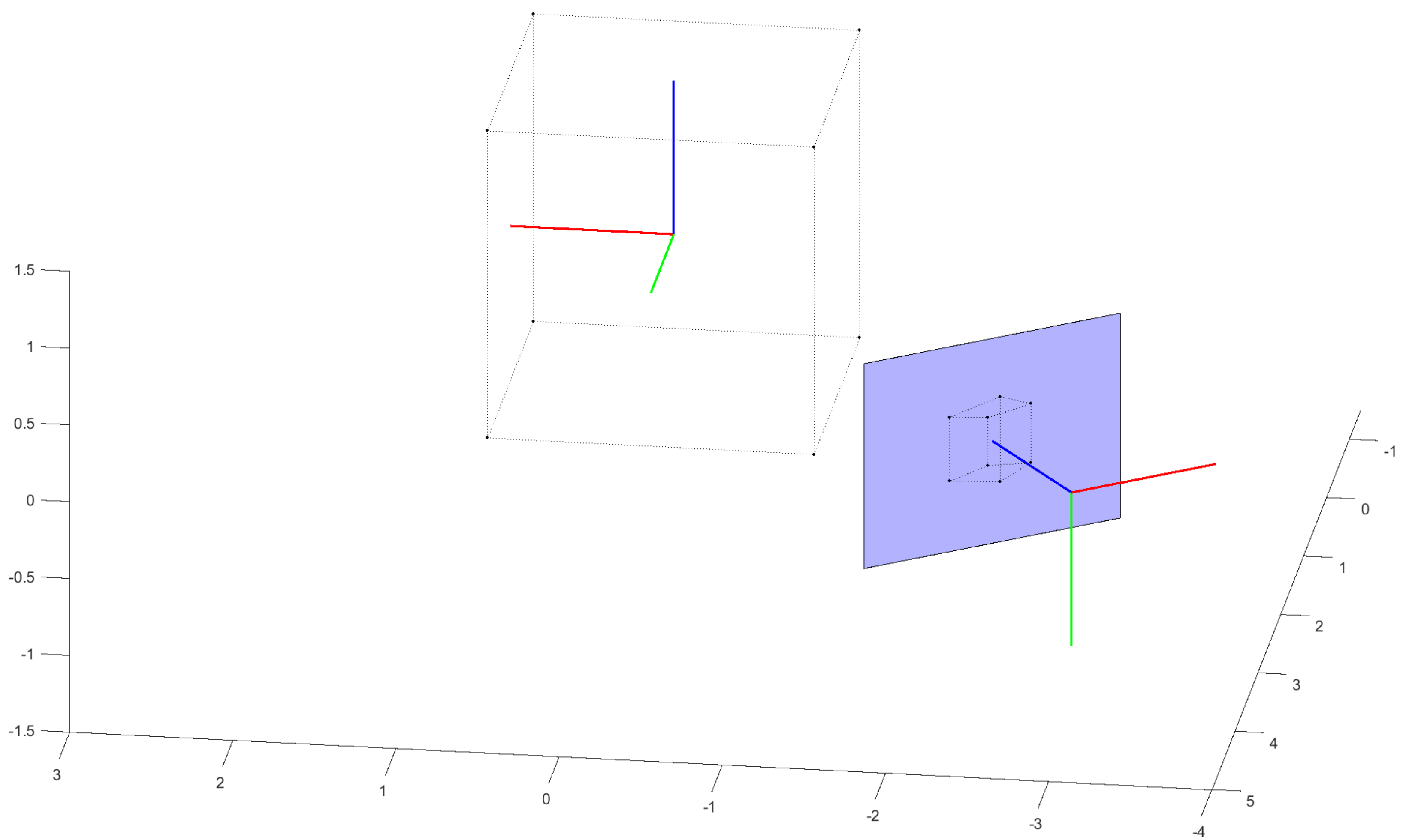
Example



Example



Example

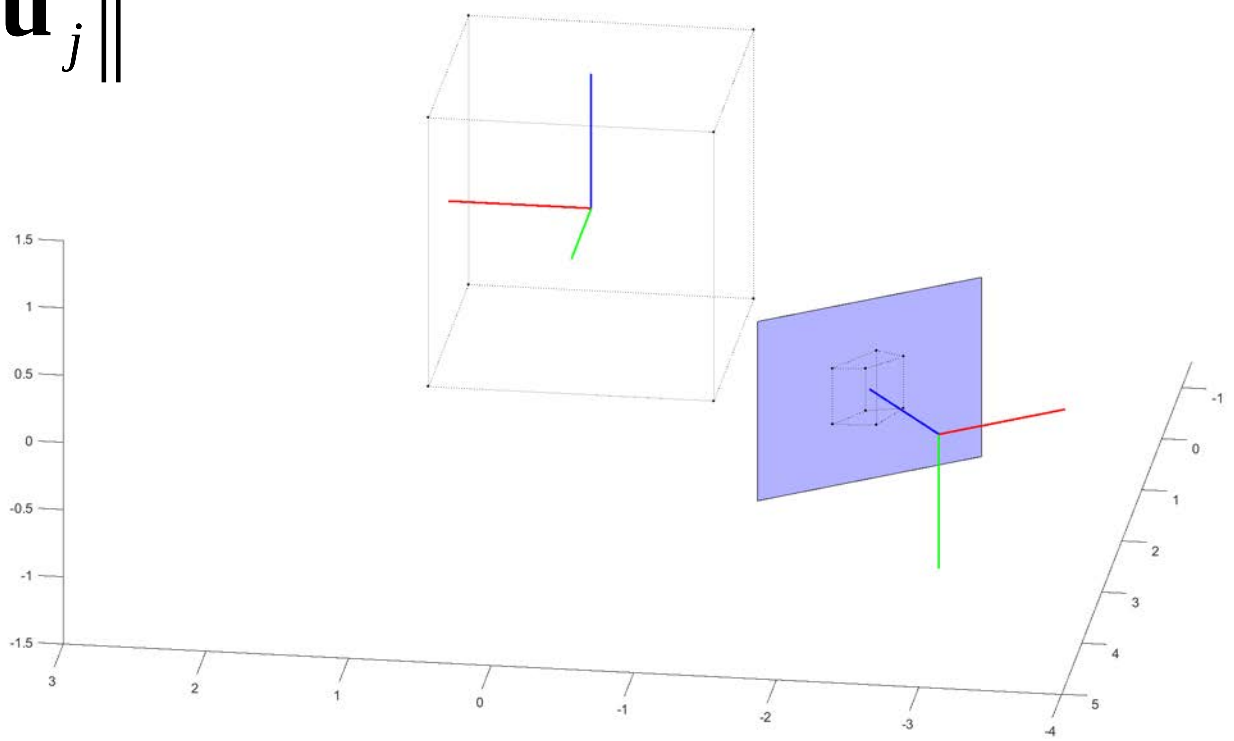


Pose estimation by minimizing reprojection error

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

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

$$\mathbf{T}_{wc}^{*} = \underset{\mathbf{T}_{wc}}{\operatorname{argmin}} \sum_{j} \left\| \pi(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_{j}^{w}) - \mathbf{u}_{j} \right\|^{2}$$

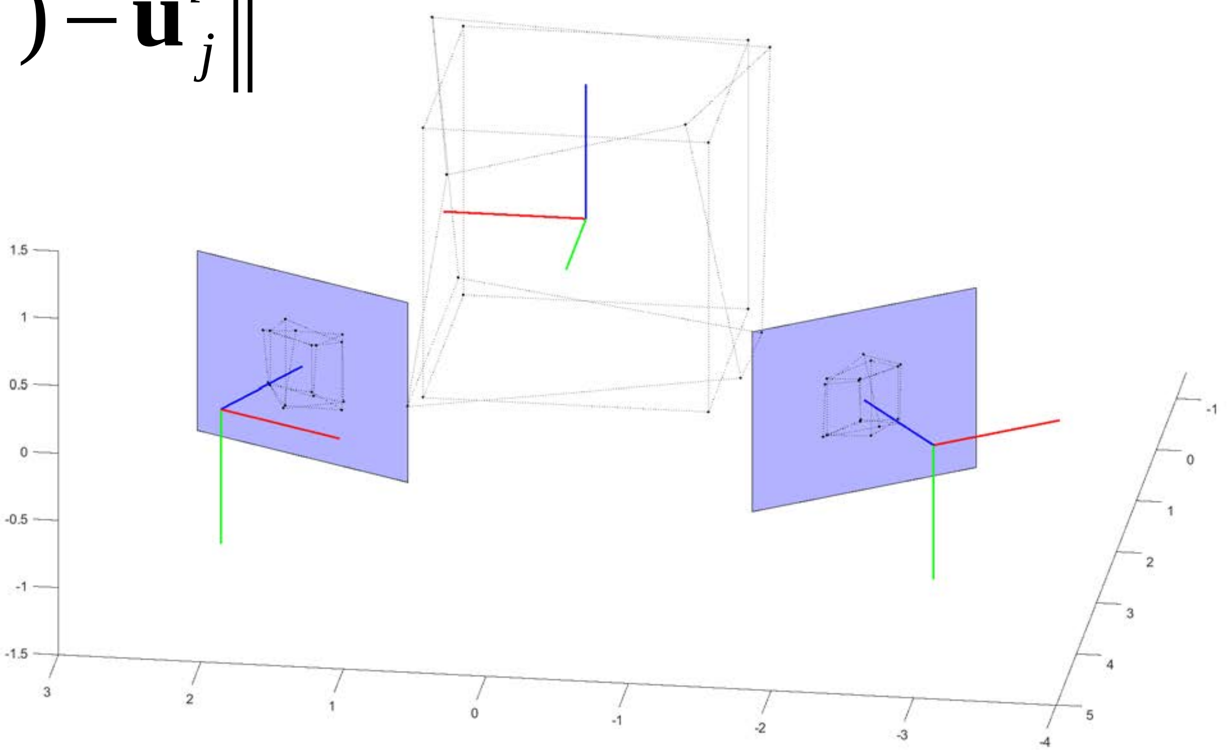


Triangulation by minimizing reprojection error

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

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

$$\mathbf{x}_j^{w*} = \underset{\mathbf{x}_j^{w*}}{\operatorname{argmin}} \sum_i \sum_j \left\| \pi_i(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{u}_j^i \right\|^2$$



Triangulation by minimizing reprojection error

Given:

- Camera poses $\mathbf{T}_{wc_i}$

Measurements:

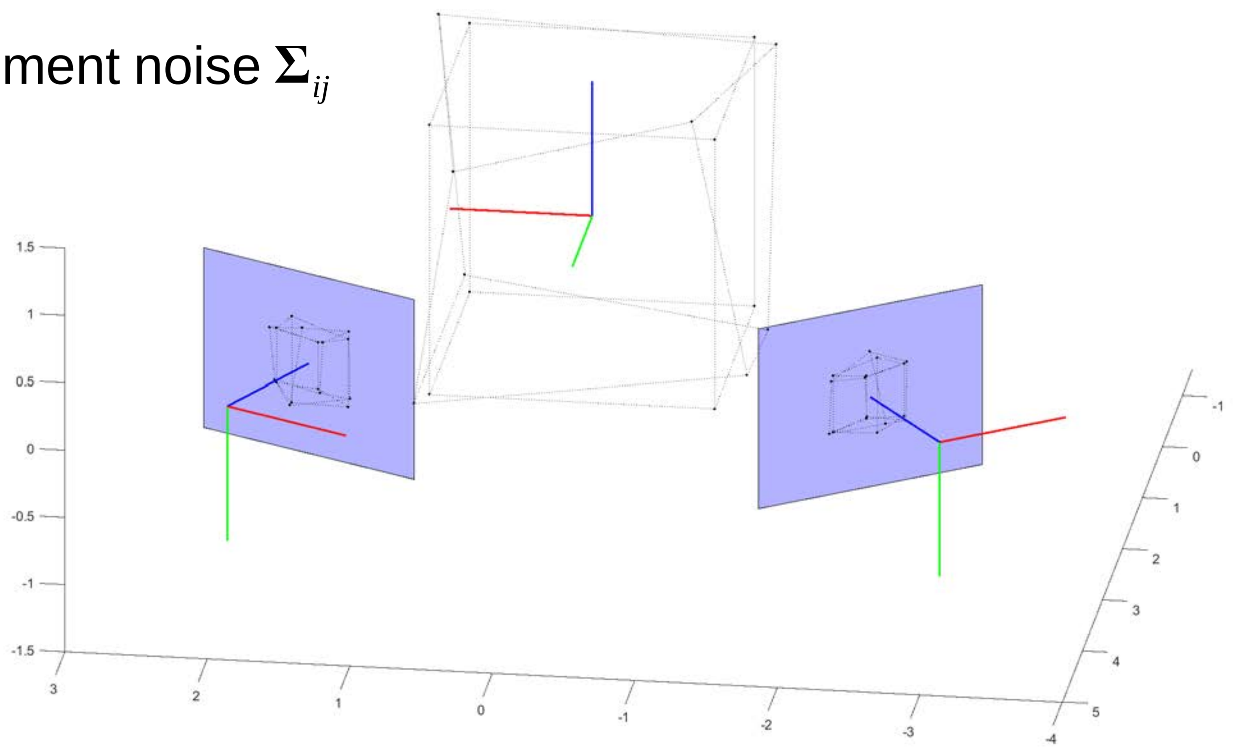
- Correspondences $\mathbf{u}_j^i \leftrightarrow \mathbf{x}_j^w$ with measurement noise $\mathbf{\Sigma}_{ij}$

State we wish to estimate:

- World points $\mathbf{x}_j^w$

Initial estimate:

- Triangulation



Applying the MAP framework

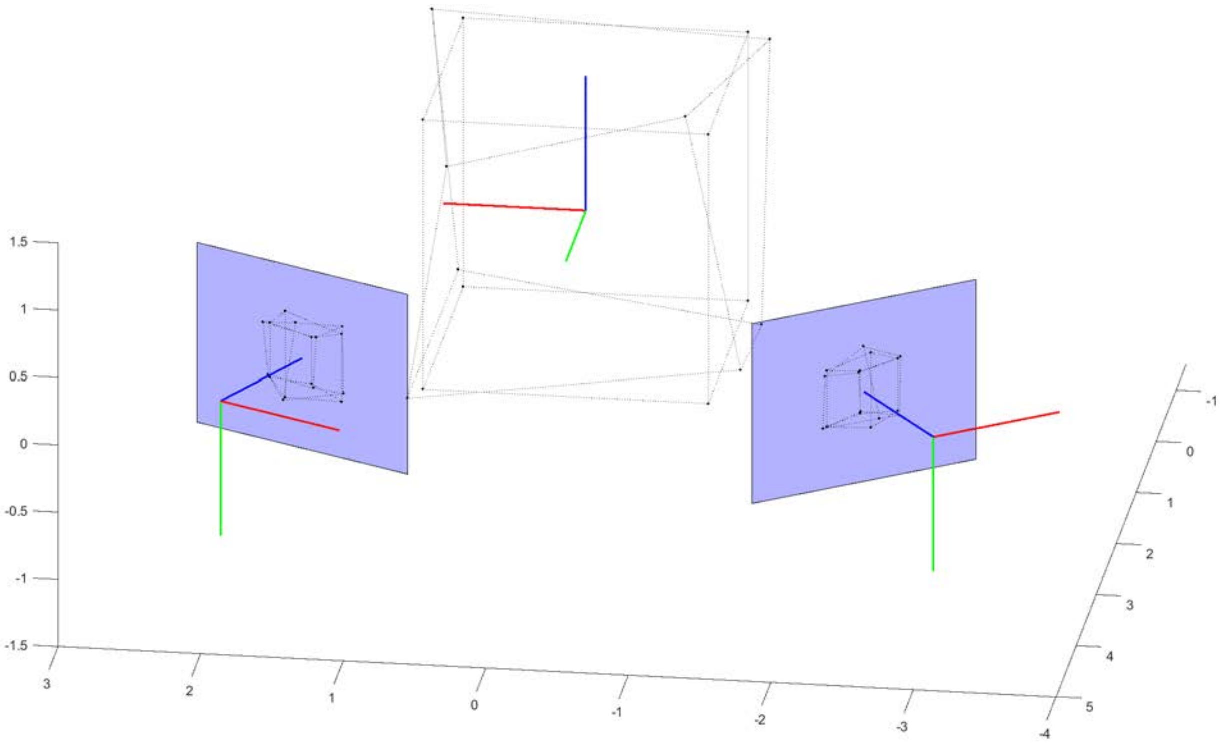
For simplicity,
we pre-calibrate to normalized image coordinates (and propagate the noise)

This gives us the measurement prediction function

$$h_{ij}(\mathbf{x}_j^w) = \pi_n(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w)$$

and measurement error function

$$e_{ij}(\mathbf{x}_j^w) = \pi_n(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{x}_{n\,j}^i$$



Applying the MAP framework

The measurement Jacobian is given by

$$
\begin{aligned}
\mathbf{J}^{h}_{\mathbf{x}^w} &= \mathbf{J}^{\pi_n(\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w)}_{\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w}\mathbf{J}^{\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w}_{\mathbf{x}^w} \\
&= \mathbf{J}^{\pi_n(\mathbf{x}^c)}_{\mathbf{x}^c}\mathbf{J}^{\mathbf{T}_{wc}^{-1}\cdot\mathbf{x}^w}_{\mathbf{x}^w} \\
&= \frac{1}{z^c}\begin{bmatrix} 1 & 0 & -x^c/z^c \\ 0 & 1 & -y^c/z^c \end{bmatrix}\mathbf{R}_{wc}^{\top} \\
&= d\begin{bmatrix} 1 & 0 & -x_n \\ 0 & 1 & -y_n \end{bmatrix}\mathbf{R}_{wc}^{\top},
\end{aligned}
$$

Applying the MAP framework

This results in the linearized weighted least squares problem

$$\begin{aligned}\delta\mathbf{x}^* &= \arg\min_{\delta\mathbf{x}} \sum_{i=1}^{k}\sum_{j=1}^{n} \left\|\mathbf{A}_{ij}\delta\mathbf{x}_j - \mathbf{b}_{ij}\right\|^2 \\ &= \arg\min_{\delta\mathbf{x}} \left\|\mathbf{A}\delta\mathbf{x} - \mathbf{b}\right\|^2 ,\end{aligned}$$

where

$$\begin{aligned}\mathbf{A}_{ij} &= \mathbf{\Sigma}_{n\ ij}^{-1/2}\mathbf{J}_{\mathbf{x}_j^w}^{h_{ij}} \\ \mathbf{b}_{ij} &= \mathbf{\Sigma}_{n\ ij}^{-1/2}(\mathbf{x}_{n\ j}^{i} - h_{ij}(\mathbf{x}_j^w)),\end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & & \\ & \ddots & \\ & & \mathbf{A}_{1n} \\ & \vdots & \\ \mathbf{A}_{k1} & & \\ & \ddots & \\ & & \mathbf{A}_{kn} \end{bmatrix} \quad \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{x}_1 \\ \vdots \\ \delta\mathbf{x}_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{11} \\ \vdots \\ \mathbf{b}_{1n} \\ \vdots \\ \mathbf{b}_{k1} \\ \vdots \\ \mathbf{b}_{kn} \end{bmatrix} .$$

Linear least-squares

The **measurement Jacobian** $\mathbf{A}$ is now a **block sparse matrix**.

For an example with *two cameras* and *three points* we have

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & & \\ & \mathbf{A}_{12} & \\ & & \mathbf{A}_{13} \\ \mathbf{A}_{21} & & \\ & \mathbf{A}_{22} & \\ & & \mathbf{A}_{23} \end{bmatrix} \qquad \delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \delta\mathbf{x}_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{12} \\ \mathbf{b}_{13} \\ \mathbf{b}_{21} \\ \mathbf{b}_{22} \\ \mathbf{b}_{23} \end{bmatrix}$$

Applying the MAP framework

The solution can be found by solving the normal equations

$$\left(\mathbf{A}^T\mathbf{A}\right)\delta\mathbf{x}^* = \mathbf{A}^T\mathbf{b}$$

Since $\mathbf{A}$ is sparse,
a sparse solver should be used.

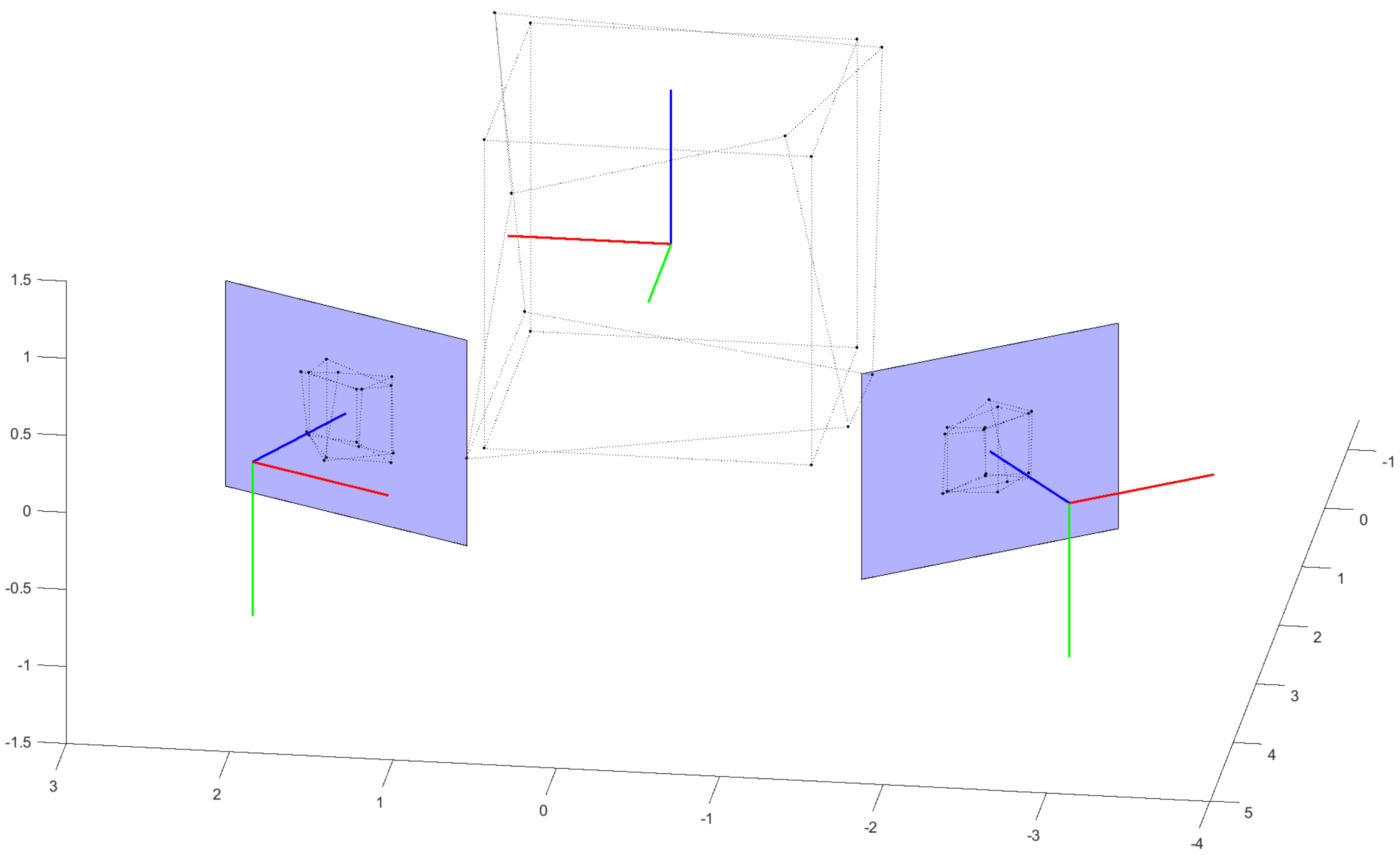
Choose a suitable inital estimate $\hat{\underline{\mathcal{X}}}^0$

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

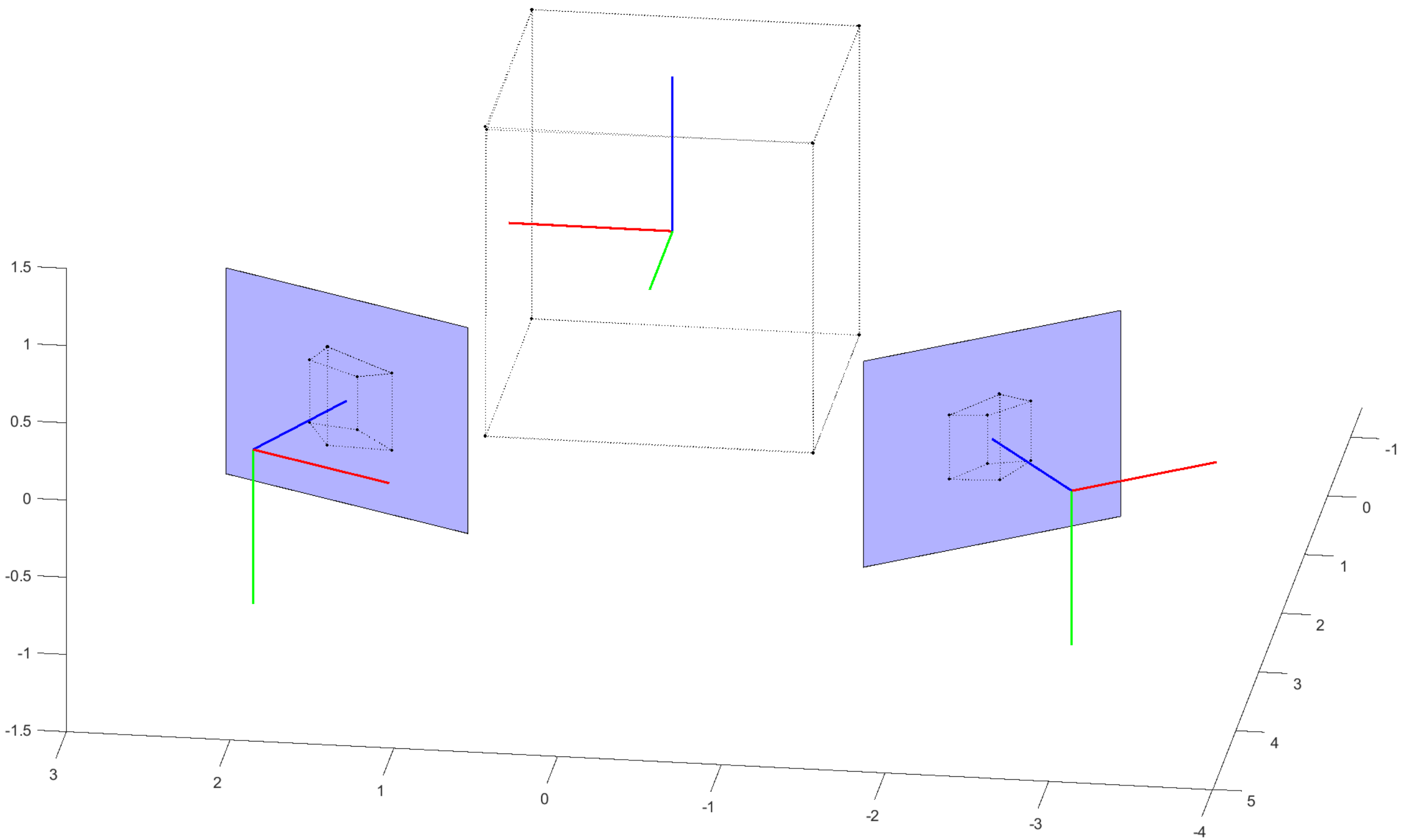
$$\underline{\boldsymbol{\tau}}^* \leftarrow \textit{Solve}\ \underset{\underline{\boldsymbol{\tau}}}{\operatorname{argmin}} \left\| \mathbf{A}\underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2$$

$$\hat{\underline{\mathcal{X}}}^{t+1} \leftarrow \hat{\underline{\mathcal{X}}}^t \oplus \underline{\boldsymbol{\tau}}^*$$

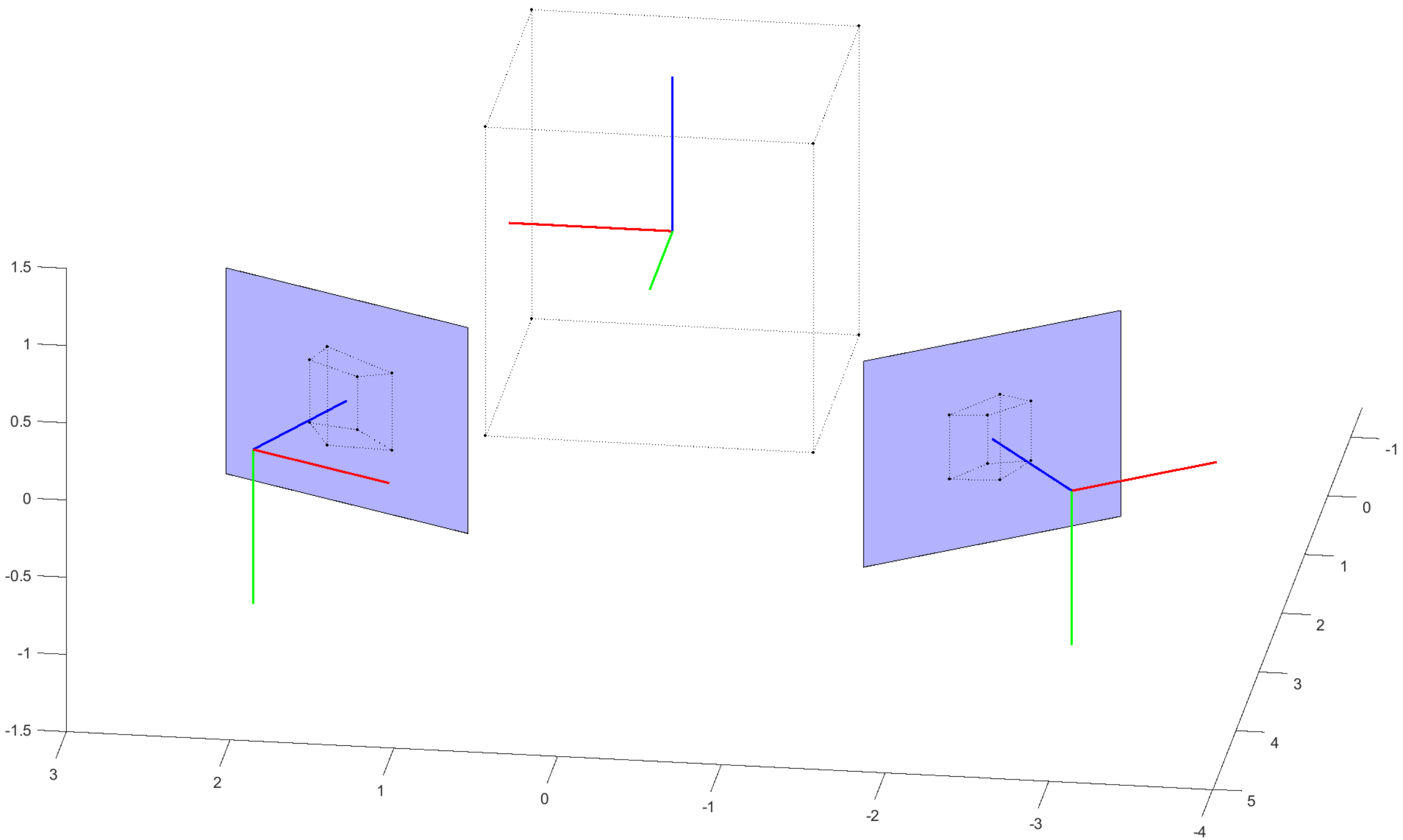
Example



Example



Example

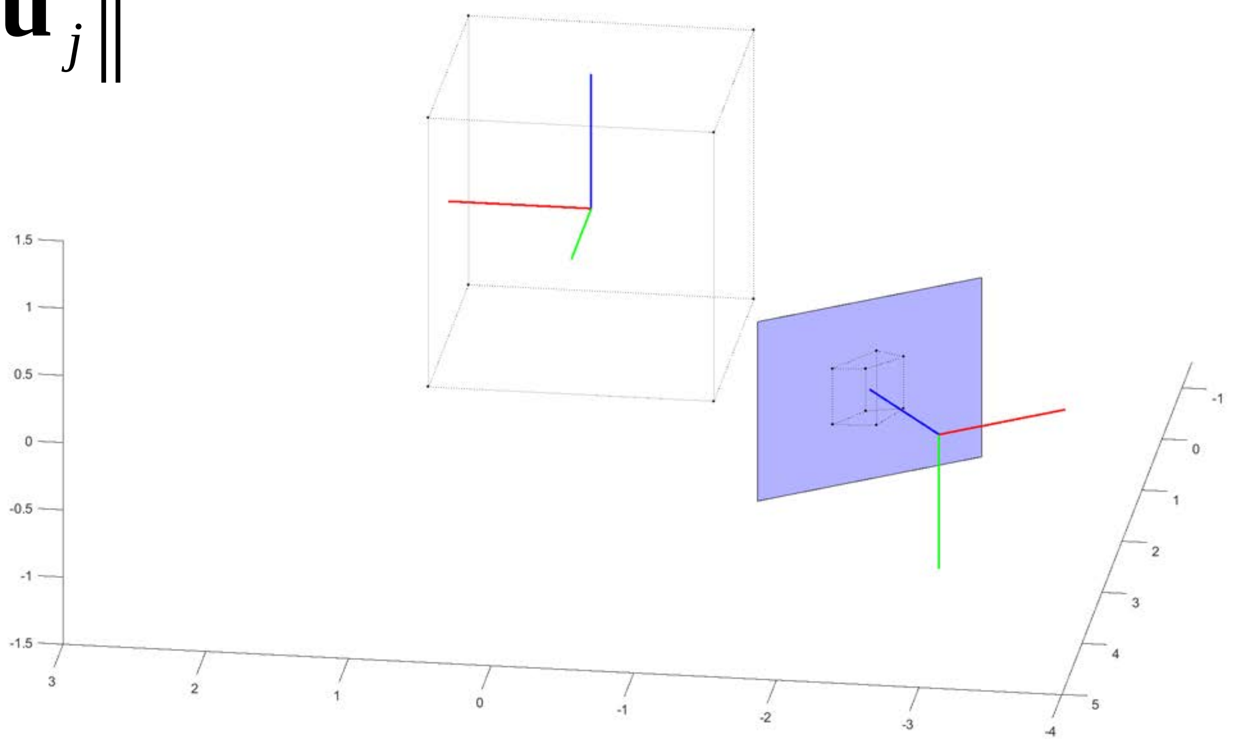


Pose estimation by minimizing reprojection error

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

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

$$\mathbf{T}_{wc}^{*} = \underset{\mathbf{T}_{wc}}{\operatorname{argmin}} \sum_{j} \left\| \pi\left(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_{j}^{w}\right) - \mathbf{u}_{j} \right\|^{2}$$

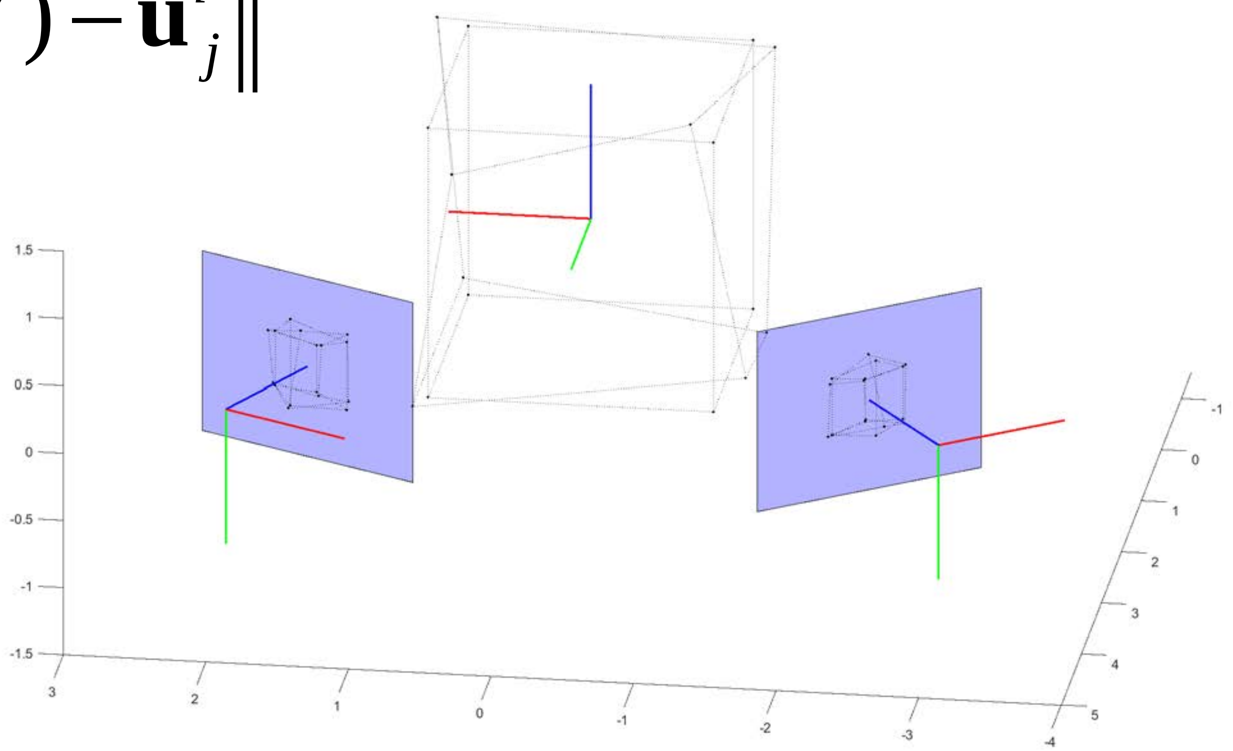


Triangulation by minimizing reprojection error

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

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

$$\mathbf{x}_j^{w*} = \underset{\mathbf{x}_j^{w*}}{\operatorname{argmin}} \sum_i \sum_j \left\| \pi_i (\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{u}_j^i \right\|^2$$

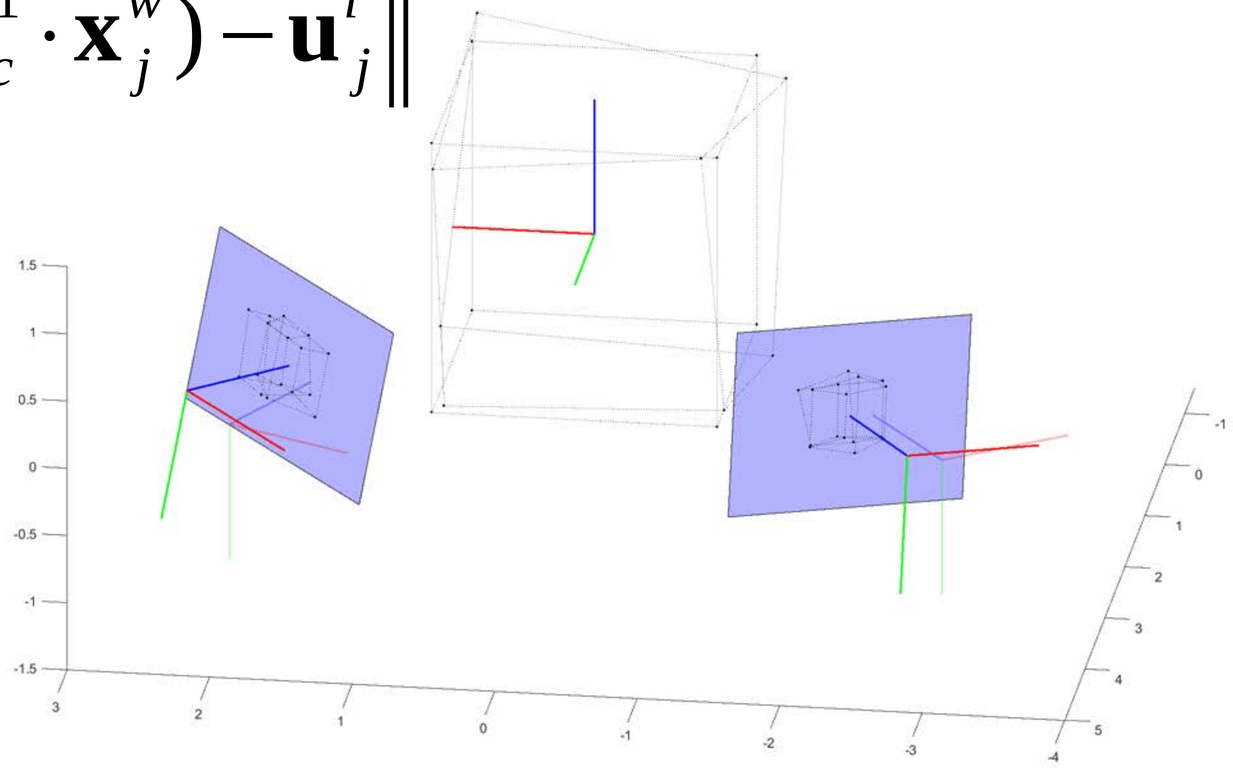


Pose and structure estimation by minimizing reprojection error

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

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

$$\left\{\mathbf{T}_{wc_i}^{*}, \mathbf{x}_j^{w*}\right\} = \underset{\mathbf{T}_{wc_i}, \mathbf{x}_j^{w}}{\operatorname{argmin}} \sum_i \sum_j \left\| \pi_i\left(\mathbf{T}_{wc}^{-1} \cdot \mathbf{x}_j^{w}\right) - \mathbf{u}_j^{i} \right\|^2$$



Pose and structure estimation by minimizing reprojection error

Given:

–

Measurements:

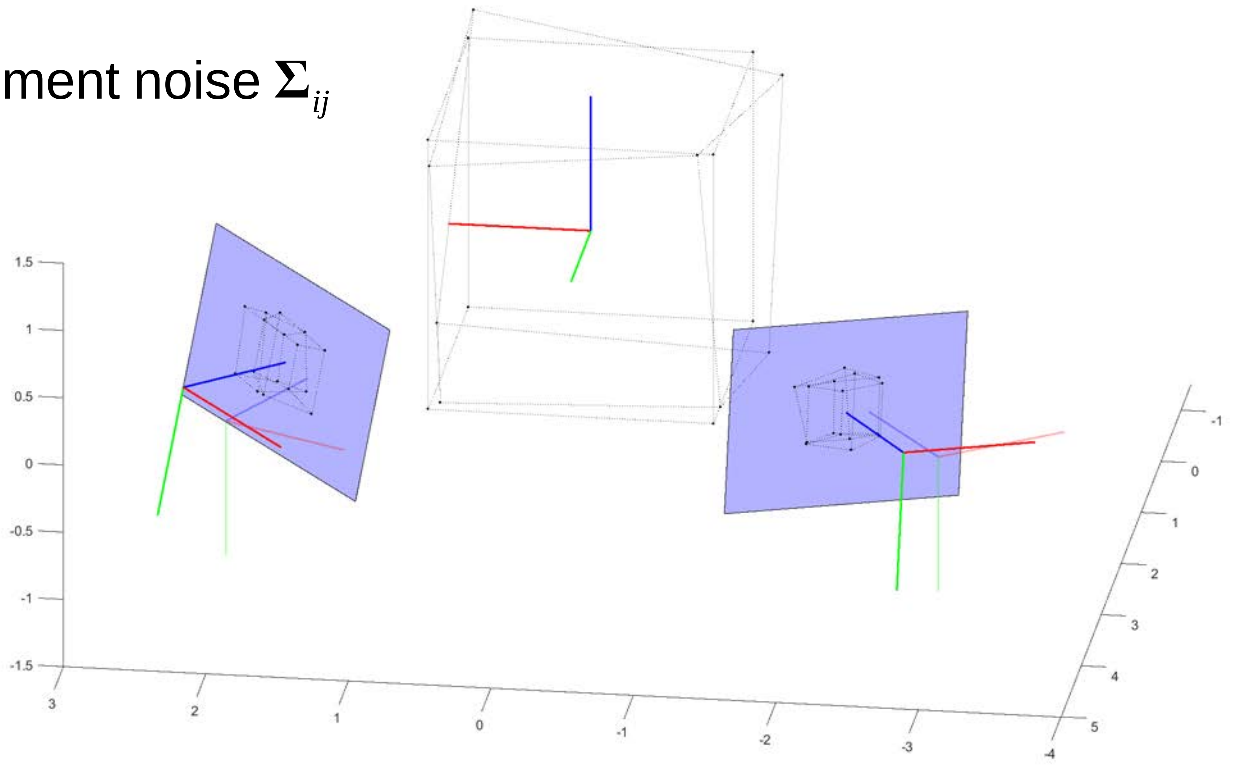
– Correspondences $\mathbf{u}_j^i \leftrightarrow \mathbf{x}_j^w$ with measurement noise $\boldsymbol{\Sigma}_{ij}$

State we wish to estimate:

– Camera poses $\mathbf{T}_{wc_i}$ *and* world points $\mathbf{x}_j^w$

Initial estimate:

– From the essential matrix (5-point algorithm)



Applying the MAP framework

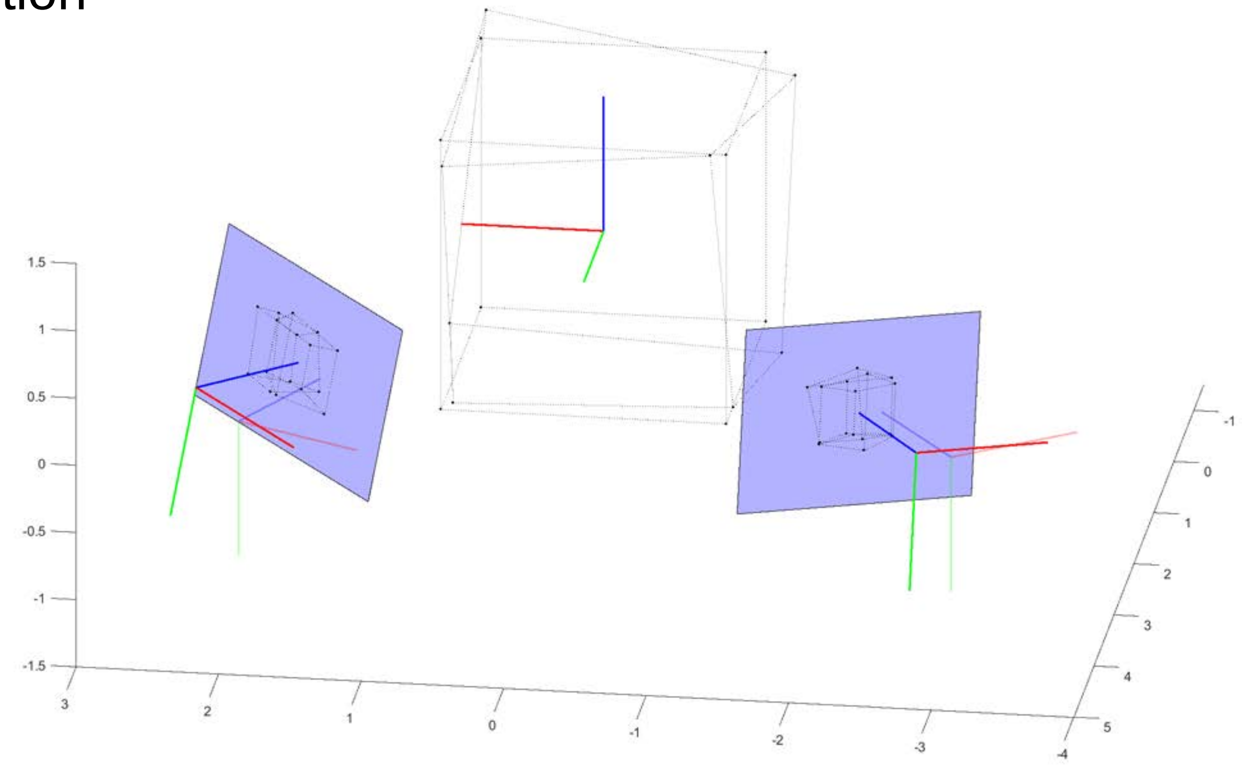
For simplicity,
we pre-calibrate to normalized image coordinates (and propagate the noise)

This gives us the measurement prediction function

$$h_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w) = \pi_n(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w)$$

and measurement error function

$$e_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w) = \pi_n(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{x}_{n\,j}^{i}$$



Applying the MAP framework

Since the measurement prediction function is a function of two variables,
we linearize it at the current state estimates as

$$
\begin{aligned}
h_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w) &= h_{ij}(\hat{\mathbf{T}}_{wc_i} \oplus \boldsymbol{\xi}_i, \hat{\mathbf{x}}_j^w + \delta\mathbf{x}_j) \\
&\approx h_{ij}(\hat{\mathbf{T}}_{wc_i}, \hat{\mathbf{x}}_j^w) + \mathbf{J}_{\hat{\mathbf{T}}_{wc_i}}^{h_{ij}} \boldsymbol{\xi}_i + \mathbf{J}_{\hat{\mathbf{x}}_j^w}^{h_{ij}} \delta\mathbf{x}_j
\end{aligned}
$$

The measurement Jacobians are given in motion-only BA and structure-only BA.

Applying the MAP framework

This results in the linearized weighted least squares problem

$$\begin{aligned}\boldsymbol{\tau}^* &= \arg\min_{\boldsymbol{\tau}} \sum_{i=1}^{k}\sum_{j=1}^{n} \left\|\mathbf{P}_{ij}\boldsymbol{\xi}_i + \mathbf{S}_{ij}\delta\mathbf{x}_j - \mathbf{b}_{ij}\right\|^2 \\ &= \arg\min_{\boldsymbol{\tau}} \left\|\mathbf{A}\boldsymbol{\tau} - \mathbf{b}\right\|^2,\end{aligned}$$

where

$$\begin{aligned}\mathbf{P}_{ij} &= \boldsymbol{\Sigma}_{n\,ij}^{-1/2}\mathbf{J}_{\mathbf{T}_{wc_i}}^{h_{ij}} \\ \mathbf{S}_{ij} &= \boldsymbol{\Sigma}_{n\,ij}^{-1/2}\mathbf{J}_{\mathbf{x}_j^w}^{h_{ij}} \\ \mathbf{b}_{ij} &= \boldsymbol{\Sigma}_{n\,ij}^{-1/2}(\mathbf{x}_{n\,j}^{i} - h_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w)),\end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{P}_{11} & & & \mathbf{S}_{11} & & \\ \vdots & & & & \ddots & \\ \mathbf{P}_{1n} & & & & & \mathbf{S}_{1n} \\ & \ddots & & & \vdots & \\ & & \mathbf{P}_{k1} & \mathbf{S}_{k1} & & \\ & & \vdots & & \ddots & \\ & & \mathbf{P}_{kn} & & & \mathbf{S}_{kn} \end{bmatrix} \quad \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \vdots \\ \boldsymbol{\xi}_k \\ \delta\mathbf{x}_1 \\ \vdots \\ \delta\mathbf{x}_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{11} \\ \vdots \\ \mathbf{b}_{1n} \\ \vdots \\ \mathbf{b}_{k1} \\ \vdots \\ \mathbf{b}_{kn} \end{bmatrix}.$$

Linear least-squares

The **measurement Jacobian** $\mathbf{A}$ is a **block sparse matrix**.

For an example with *two cameras* and *three points* we have

$$\mathbf{A} = \begin{bmatrix} \mathbf{P}_{11} & & \mathbf{S}_{11} & & \\ \mathbf{P}_{12} & & & \mathbf{S}_{12} & \\ \mathbf{P}_{13} & & & & \mathbf{S}_{13} \\ & \mathbf{P}_{21} & \mathbf{S}_{21} & & \\ & \mathbf{P}_{22} & & \mathbf{S}_{22} & \\ & \mathbf{P}_{23} & & & \mathbf{S}_{23} \end{bmatrix} \qquad \underline{\boldsymbol{\tau}} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \delta\mathbf{x}_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{12} \\ \mathbf{b}_{13} \\ \mathbf{b}_{21} \\ \mathbf{b}_{22} \\ \mathbf{b}_{23} \end{bmatrix}$$

Applying the MAP framework

The solution can be found by solving the normal equations

$$\left(\mathbf{A}^T \mathbf{A}\right)\underline{\boldsymbol{\tau}}^* = \mathbf{A}^T \mathbf{b}$$

Since $\mathbf{A}$ is sparse,
a sparse solver should be used.

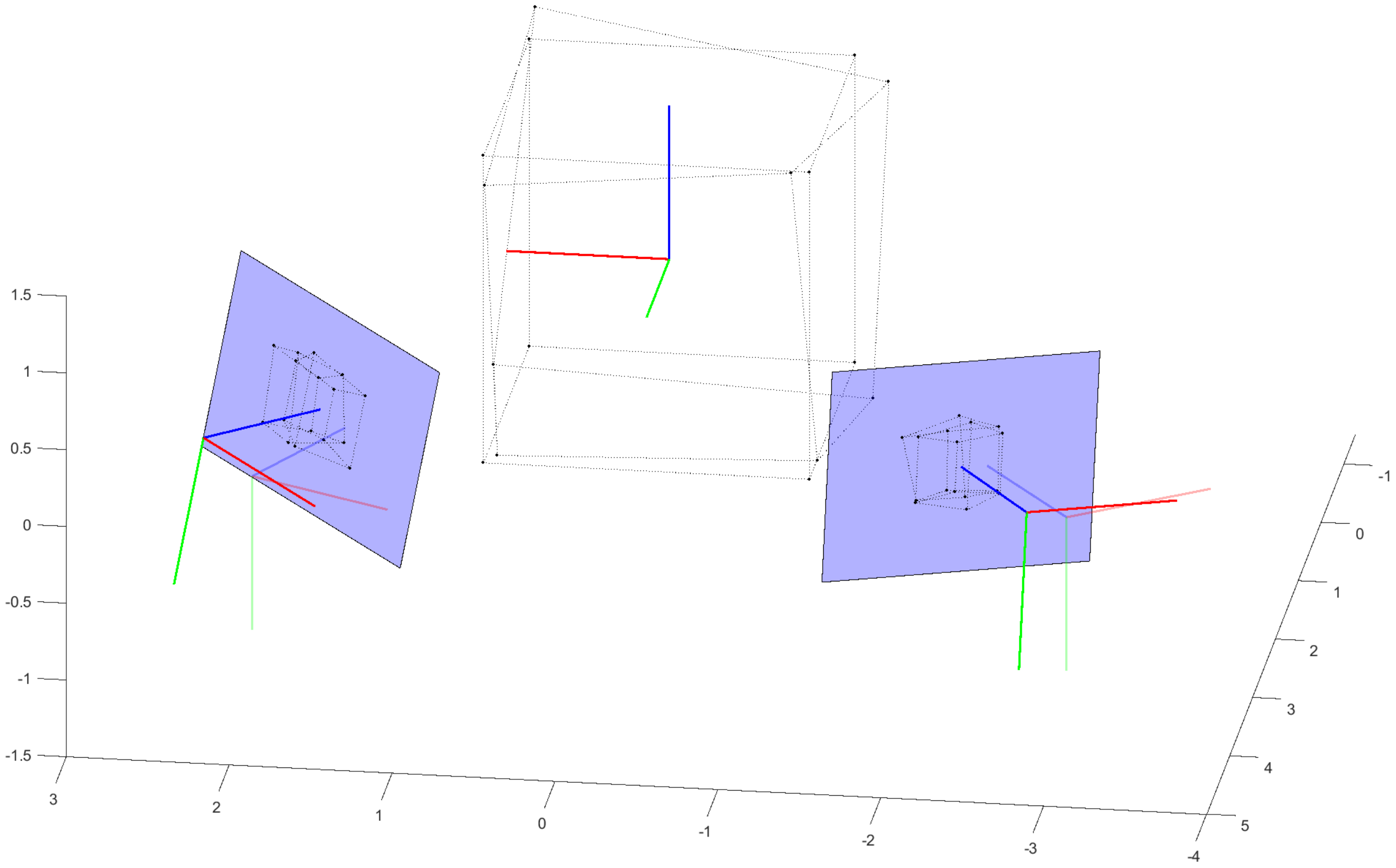
$$\text{Choose a suitable inital estimate } \hat{\underline{\mathcal{X}}}^0$$

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

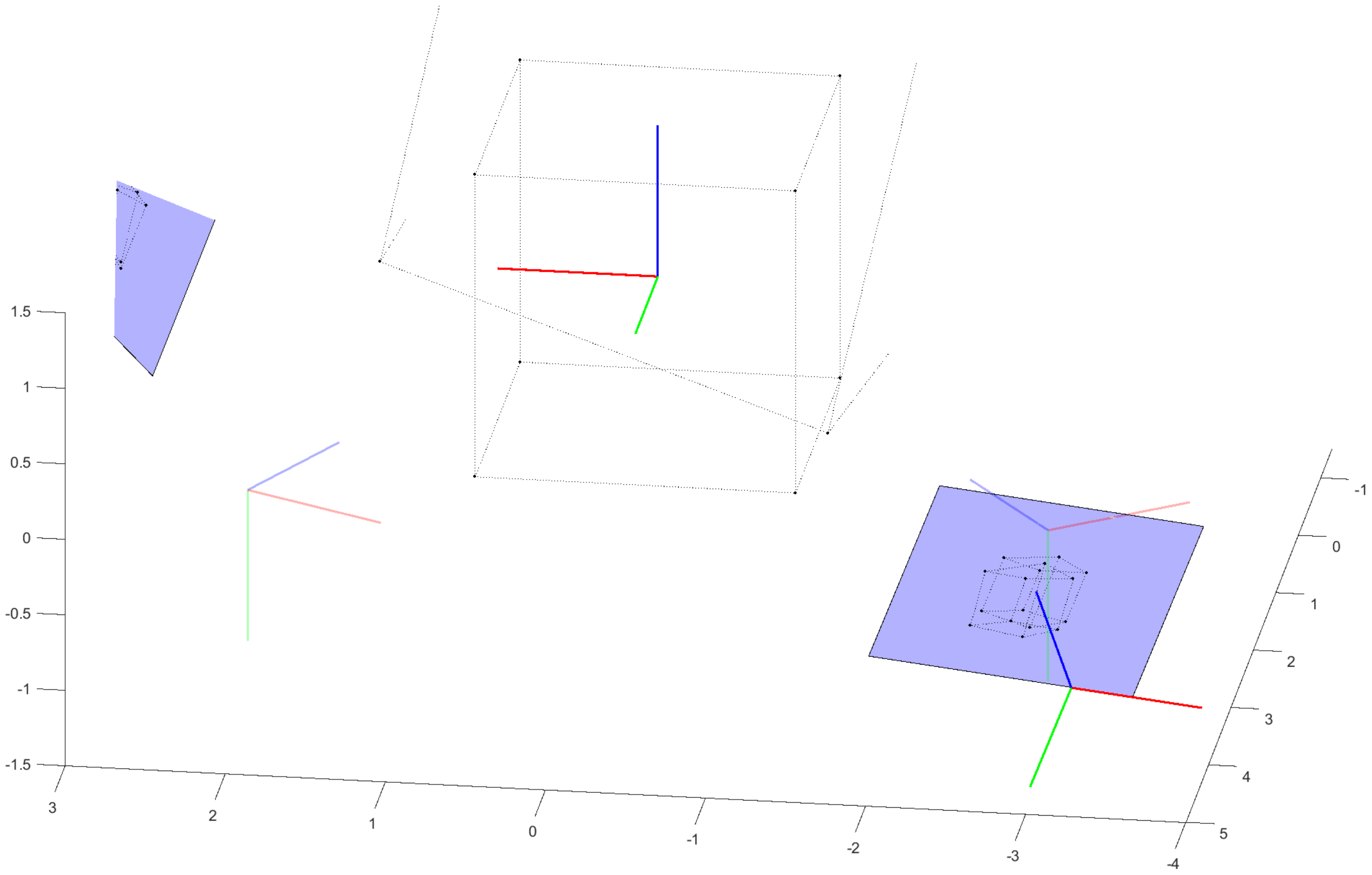
$$\underline{\boldsymbol{\tau}}^* \leftarrow \textit{Solve } \underset{\underline{\boldsymbol{\tau}}}{\operatorname{argmin}} \left\| \mathbf{A}\underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2$$

$$\hat{\underline{\mathcal{X}}}^{t+1} \leftarrow \hat{\underline{\mathcal{X}}}^t \oplus \underline{\boldsymbol{\tau}}^*$$

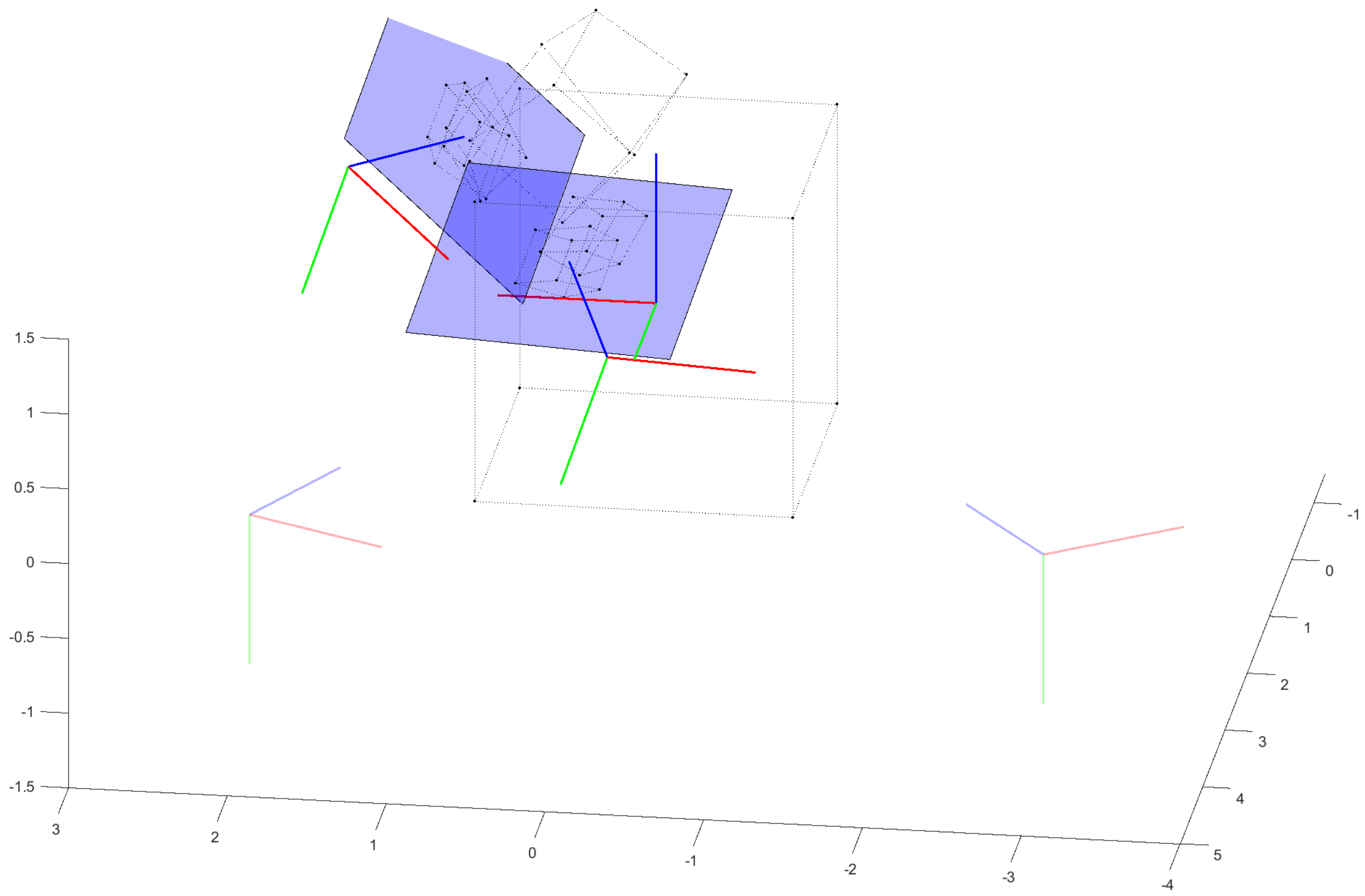
Example



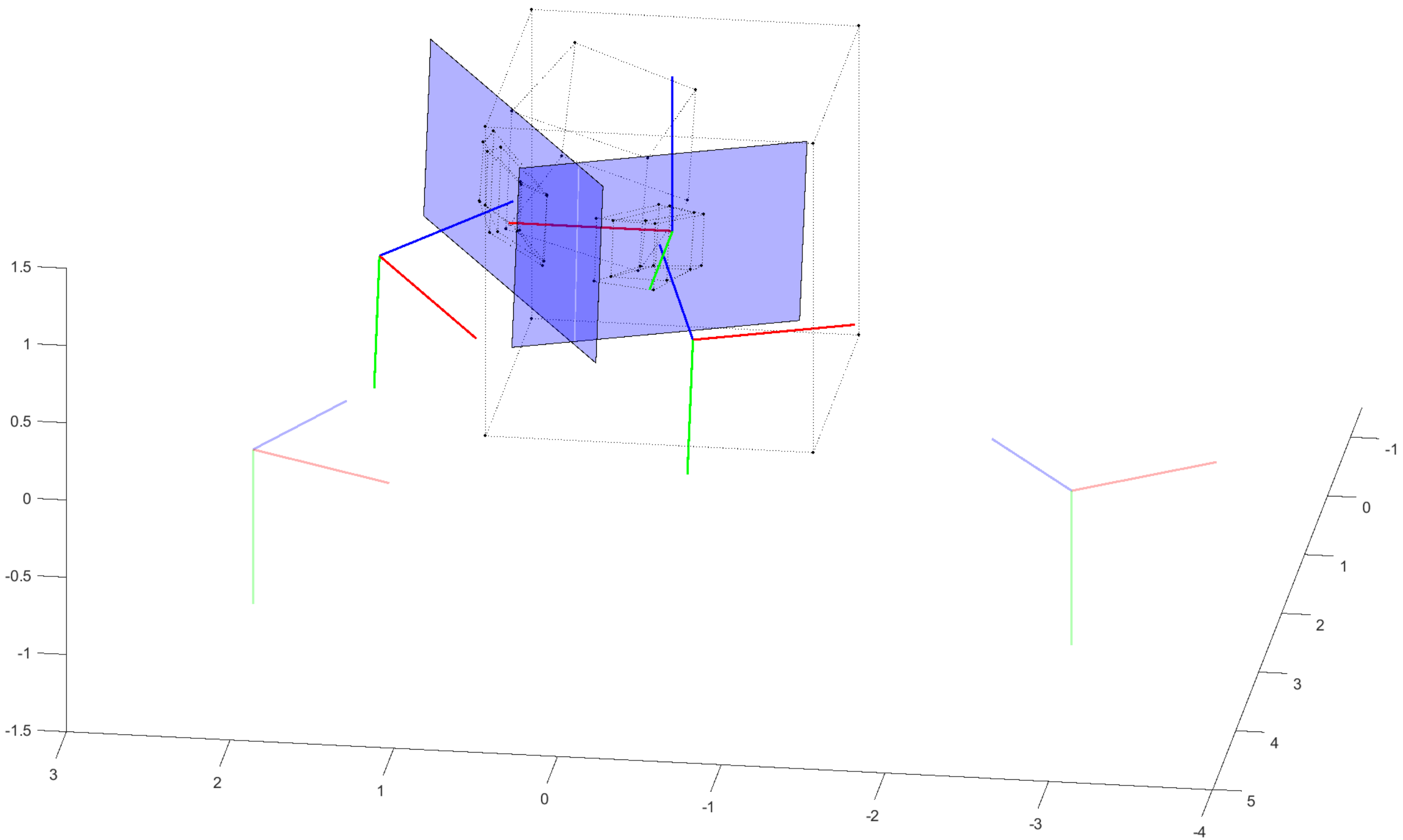
Example



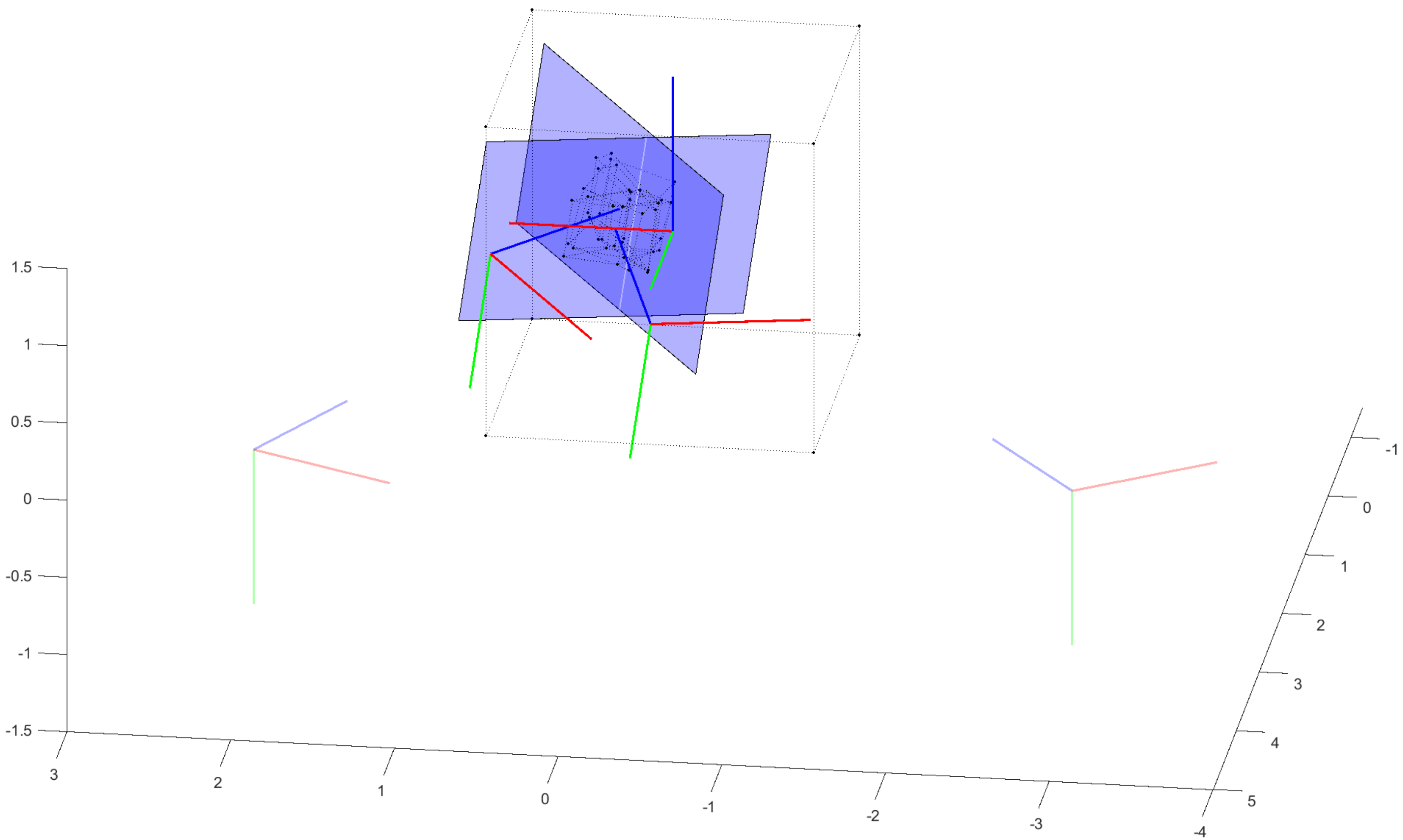
Example



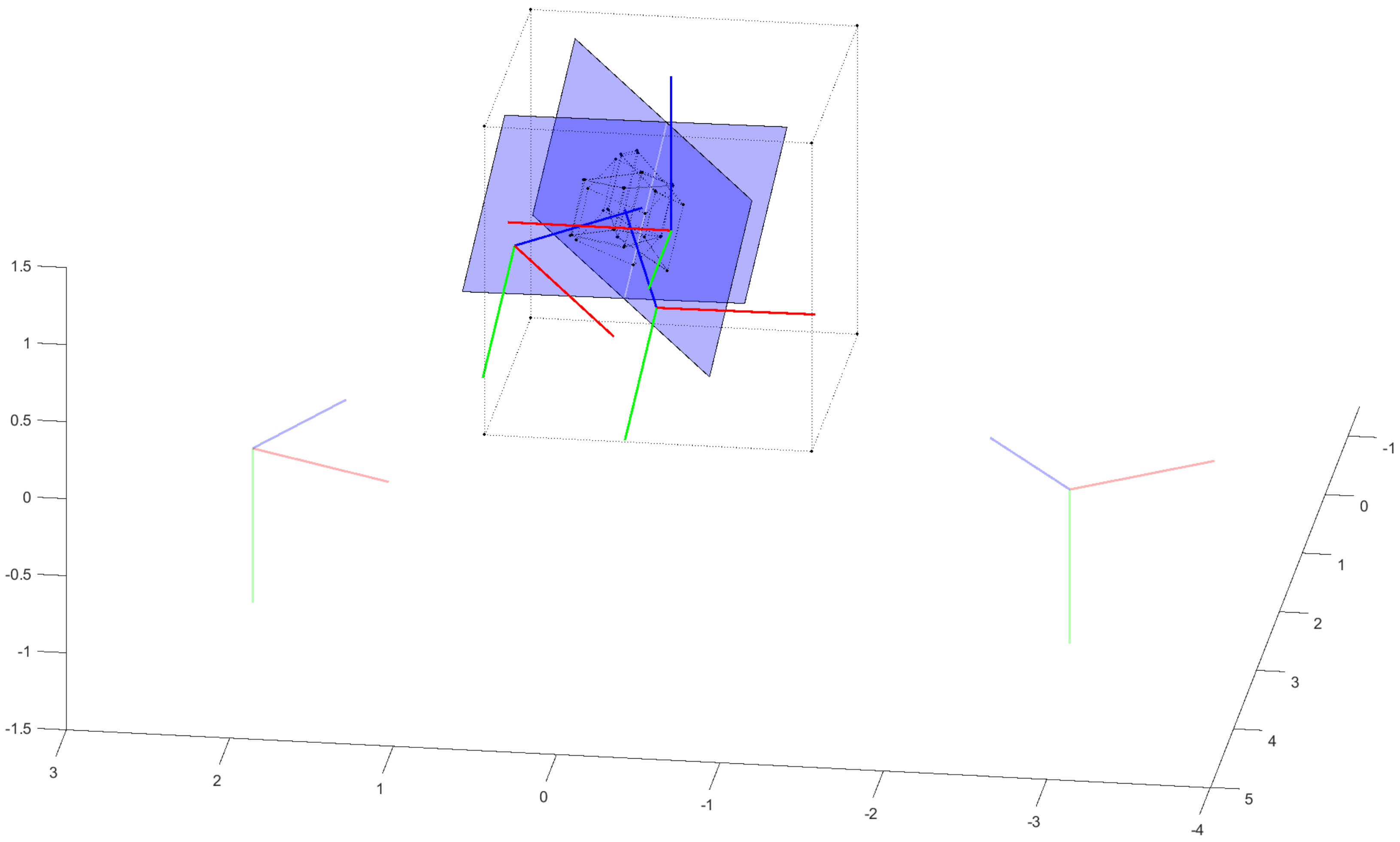
Example



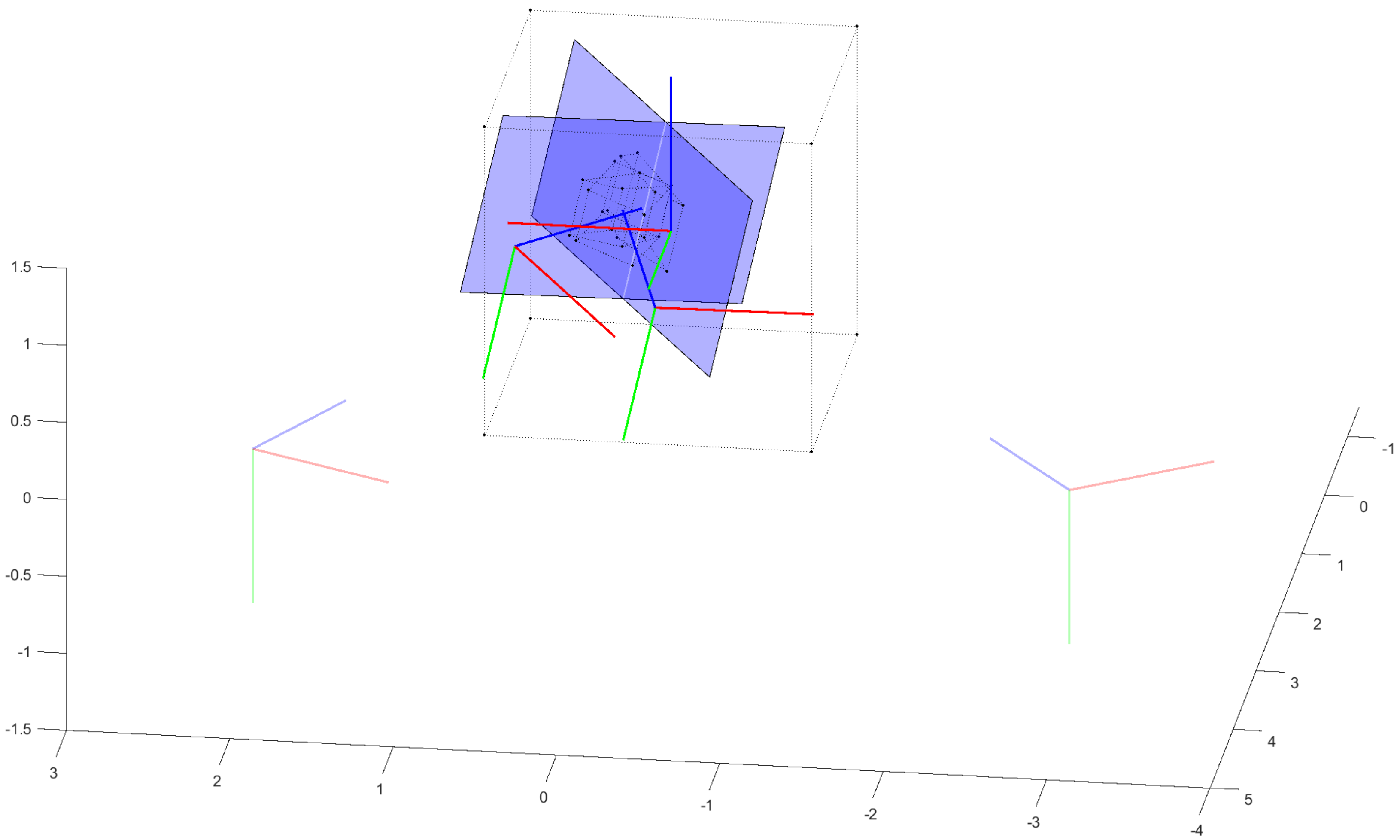
Example



Example

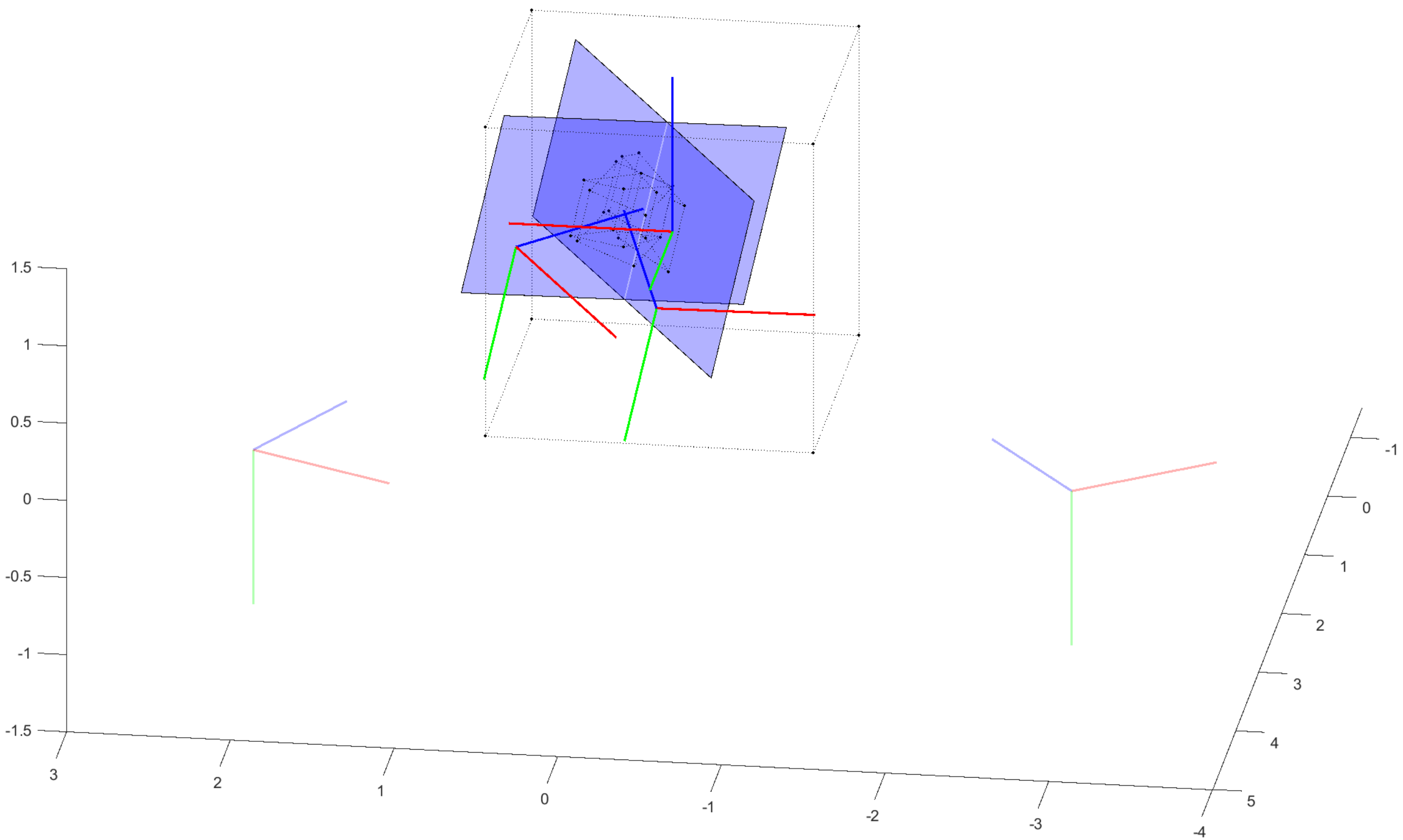


Example



Example

Why does this fail?



Gauge freedom

The solution is not uniquely determined!

- The Hessian is singular!
- We can apply any 7DOF similarity transform to the cameras without affecting the objective function

Gauge freedom

The solution is not uniquely determined!

- The Hessian is singular!
- We can apply any 7DOF similarity transform to the cameras without affecting the objective function

Possible solutions:

- ~~Use **Levenberg-Marquardt** optimization~~
- Add priors on poses and points
- Fuse with other information, such as GPS and IMU

Adding priors

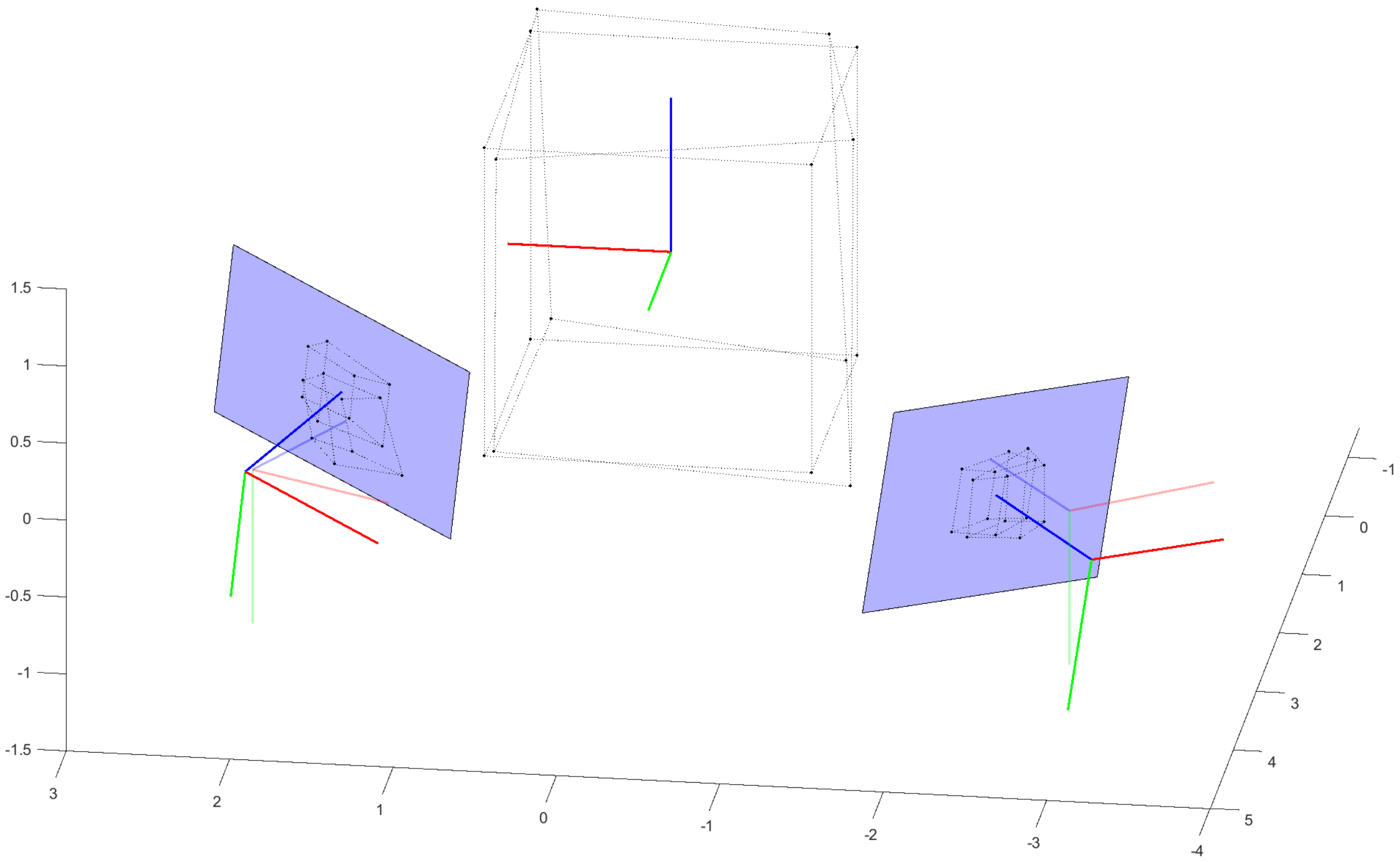
Prior on first pose and first point

$$\mathbf{A} = \begin{bmatrix} \mathbf{P}_{11} & & \mathbf{S}_{11} & & \\ \mathbf{P}_{12} & & & \mathbf{S}_{12} & \\ \mathbf{P}_{13} & & & & \mathbf{S}_{13} \\ & \mathbf{P}_{21} & \mathbf{S}_{21} & & \\ & \mathbf{P}_{22} & & \mathbf{S}_{22} & \\ & \mathbf{P}_{23} & & & \mathbf{S}_{23} \\ \mathbf{I}_{2\times 6} & & & & \\ & & \mathbf{I}_{2\times 3} & & \end{bmatrix} \qquad \underline{\boldsymbol{\tau}} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \\ \delta\mathbf{x}_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{12} \\ \mathbf{b}_{13} \\ \mathbf{b}_{21} \\ \mathbf{b}_{22} \\ \mathbf{b}_{23} \\ \mathbf{b}_{\xi_1^{prior}} \\ \mathbf{b}_{\delta\mathbf{x}_1^{prior}} \end{bmatrix}$$

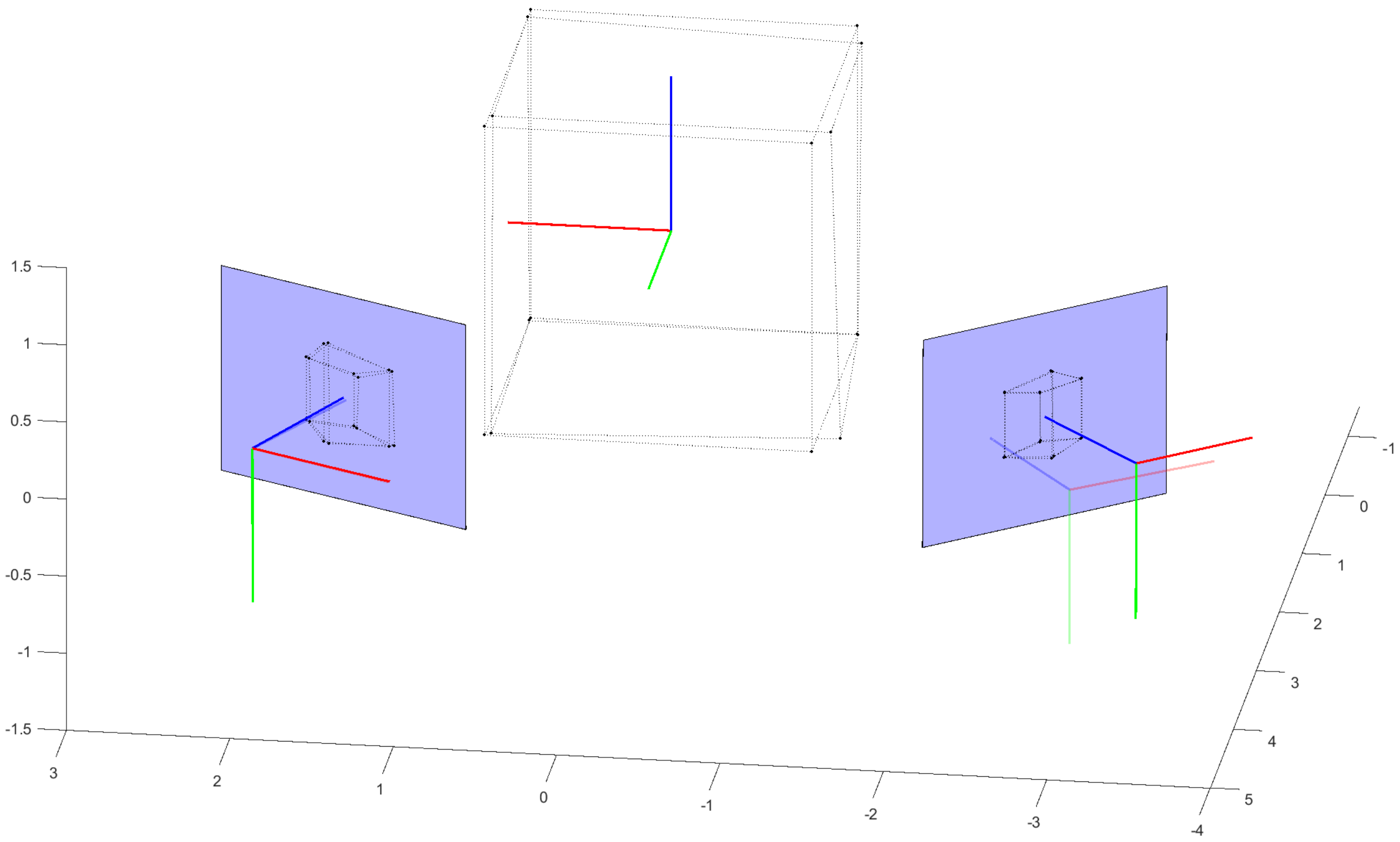
$$\mathbf{b}_{\xi_1^{prior}} = \mathbf{T}_{wc_1}^{prior} \ominus \mathbf{T}_{wc_1}$$

$$\mathbf{b}_{\delta\mathbf{x}_1^{prior}} = \mathbf{x}_1^{w,prior} - \mathbf{x}_1^{w}$$

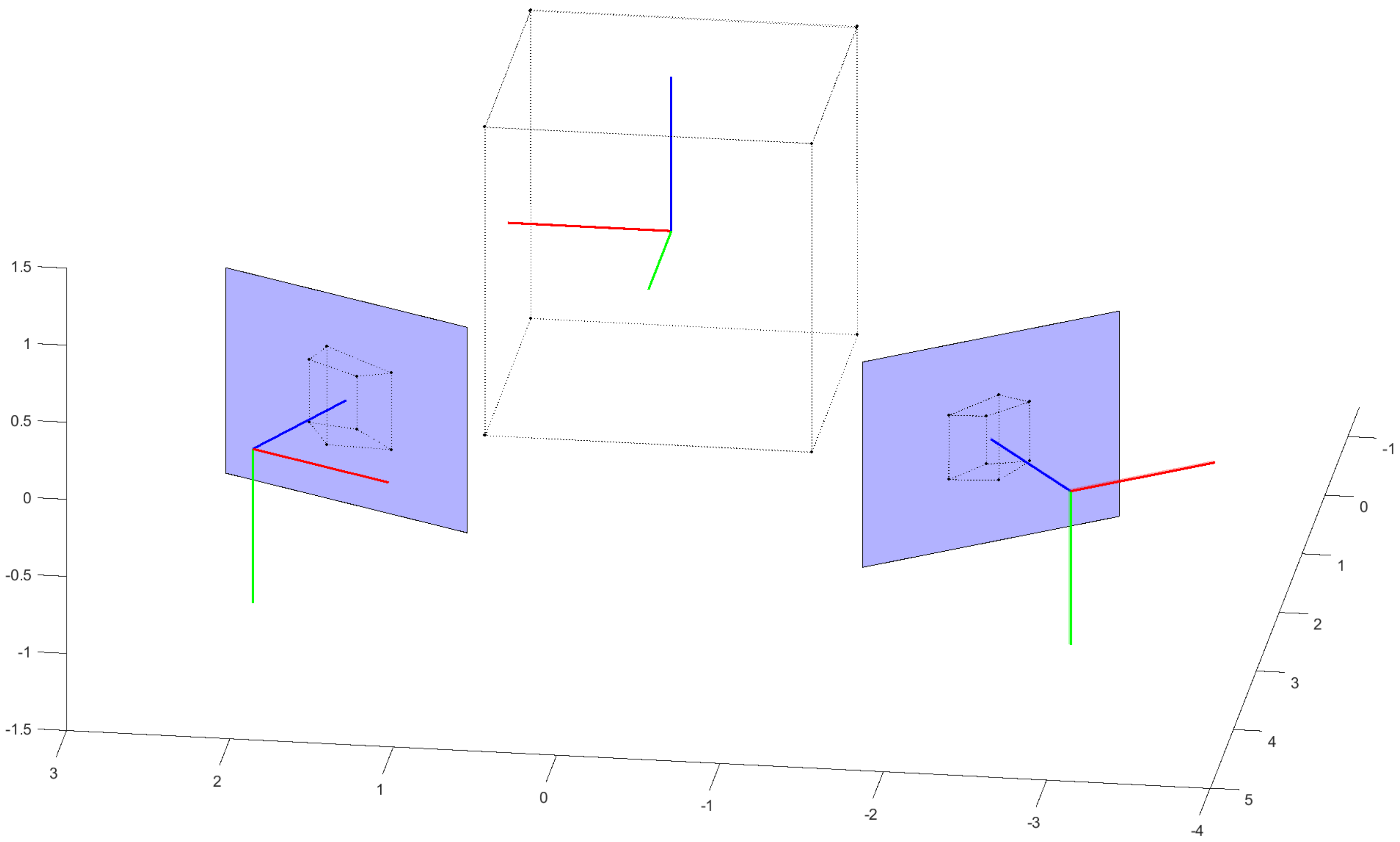
Example



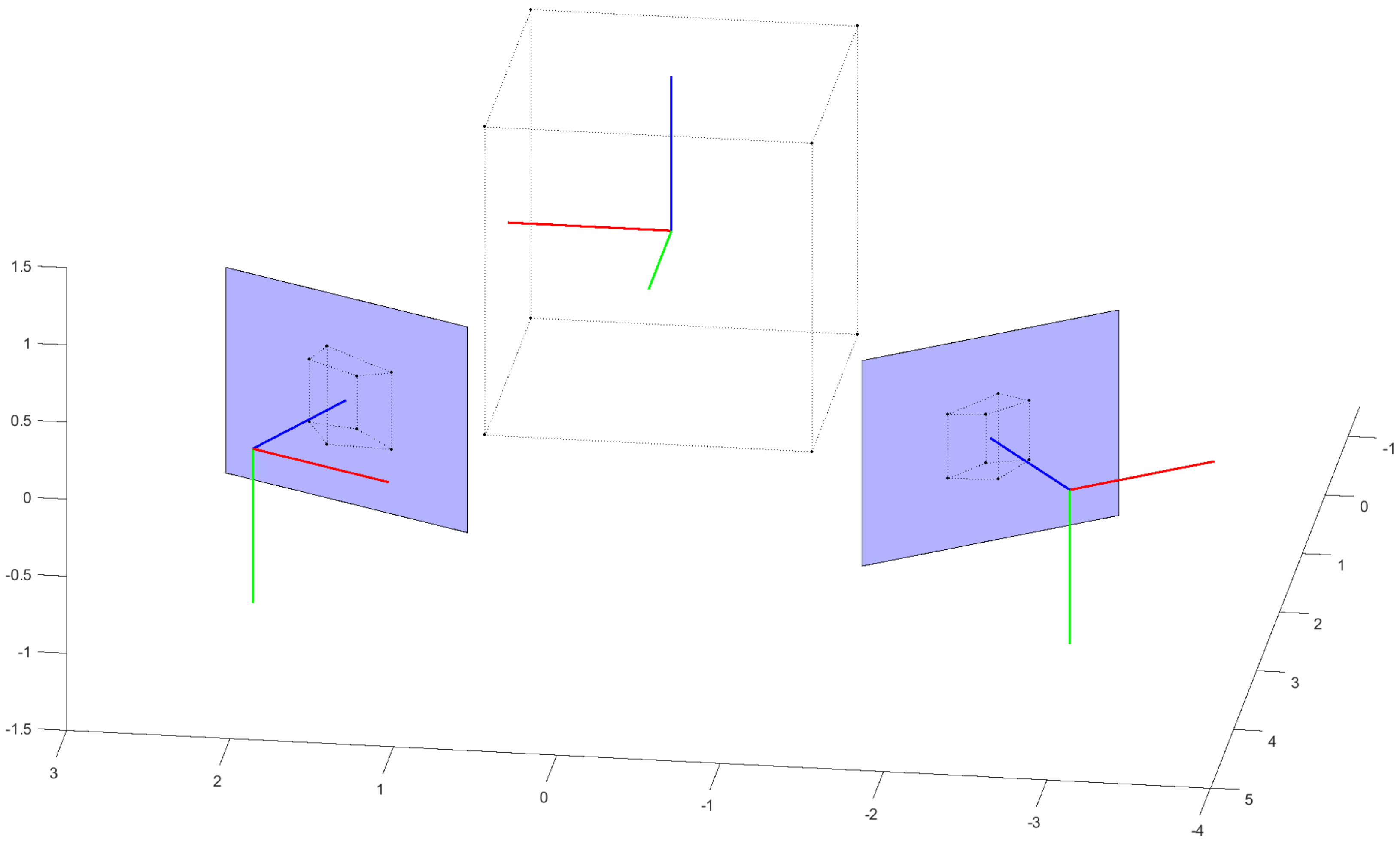
Example



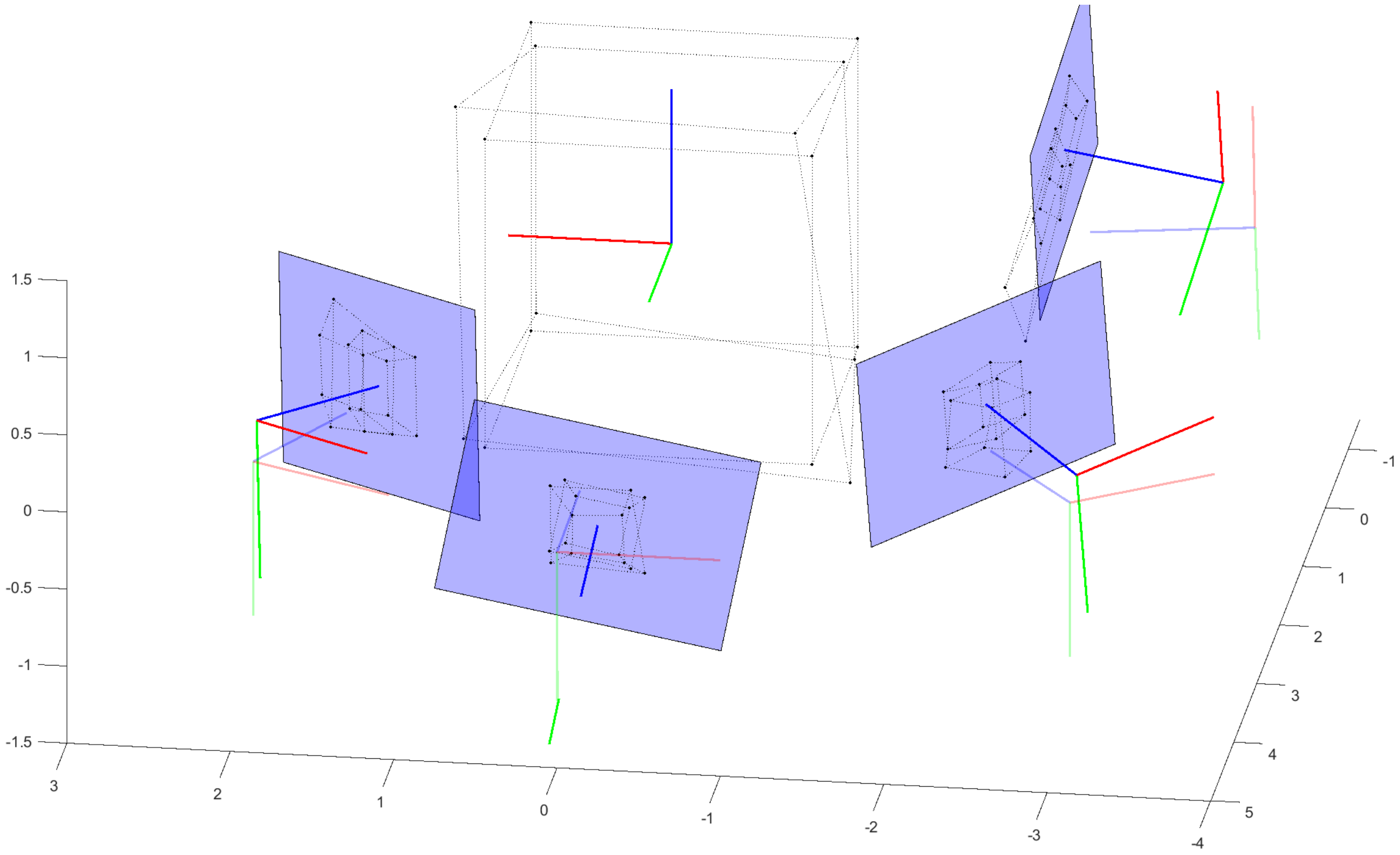
Example



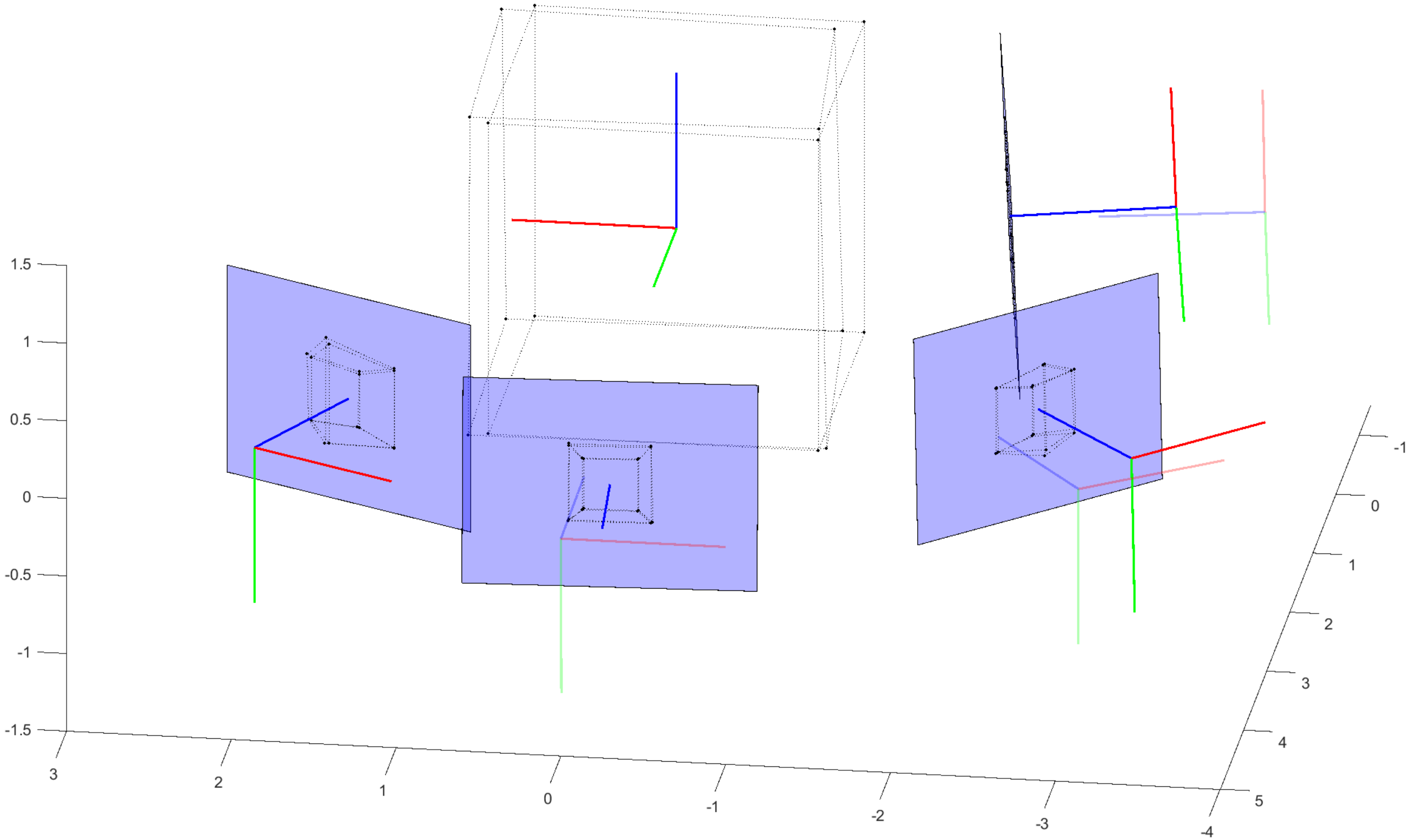
Example



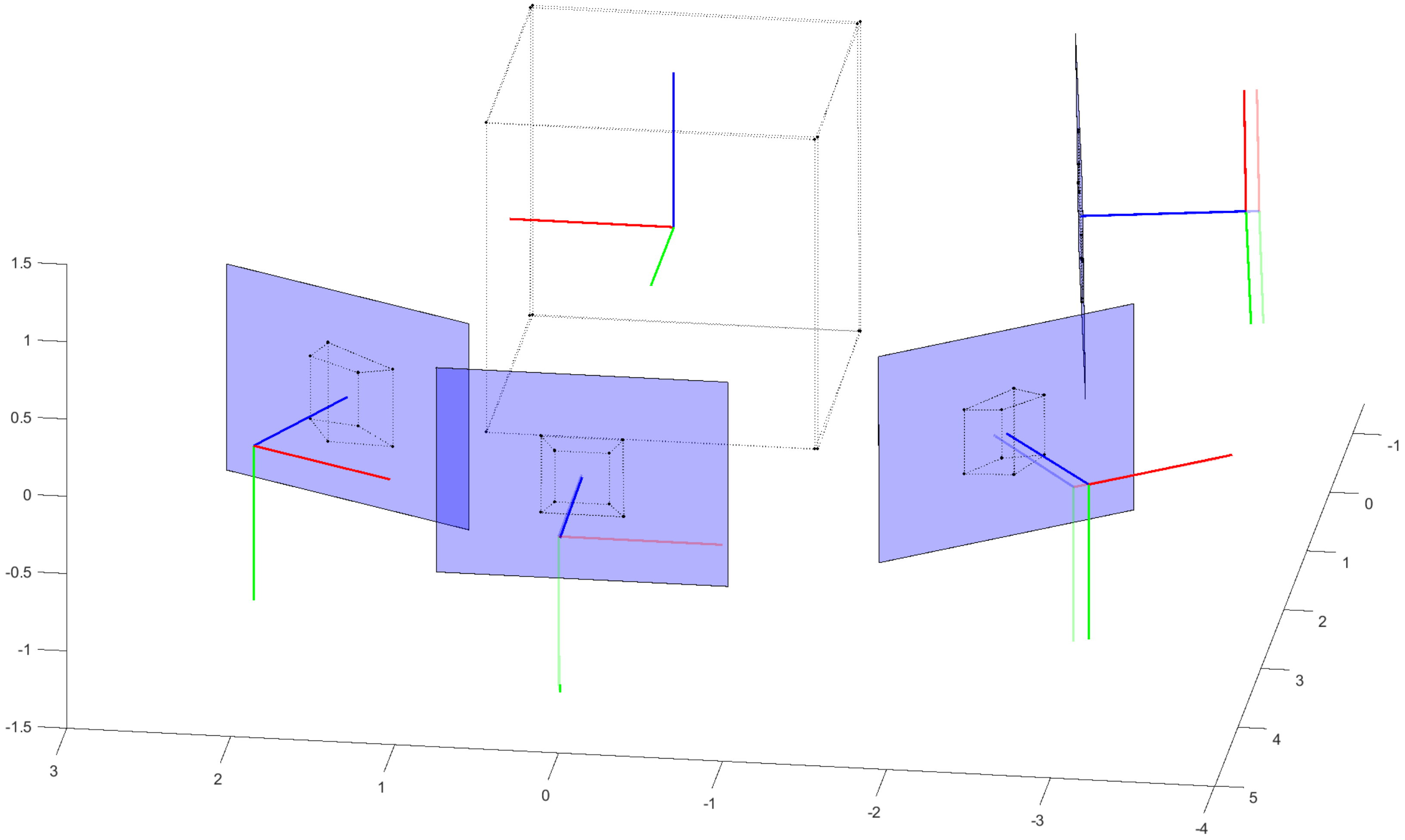
Example



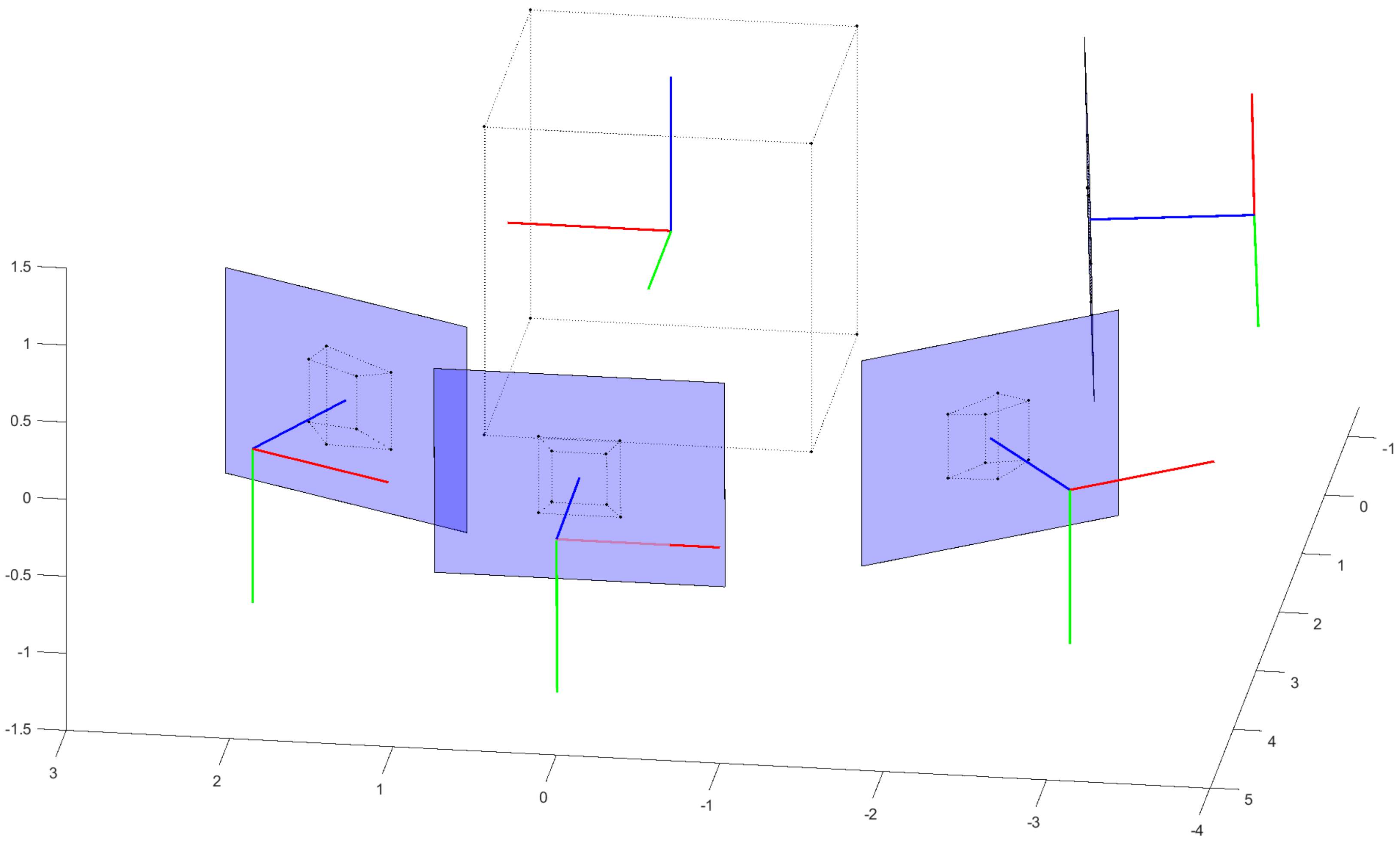
Example



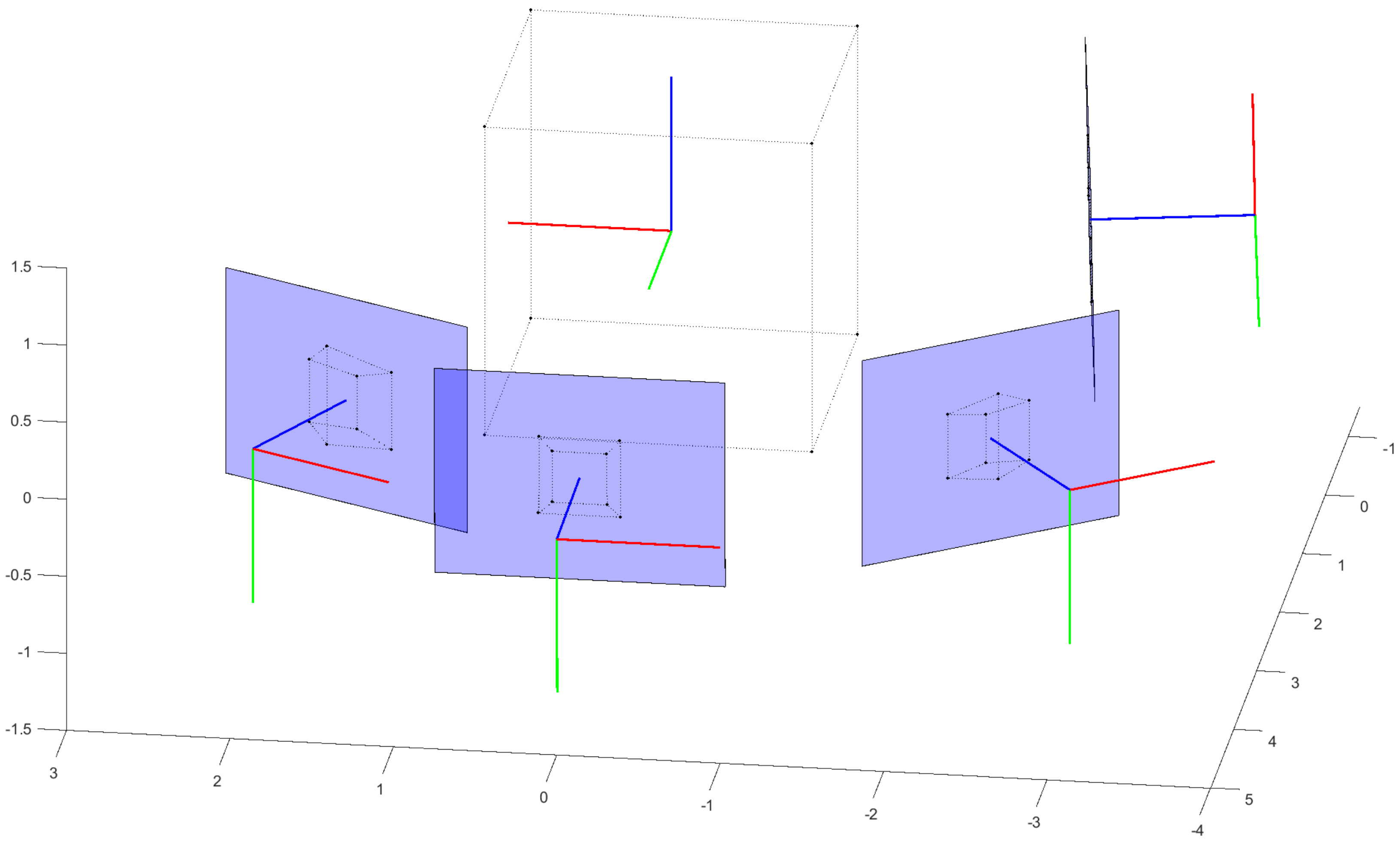
Example



Example



Example



Part III

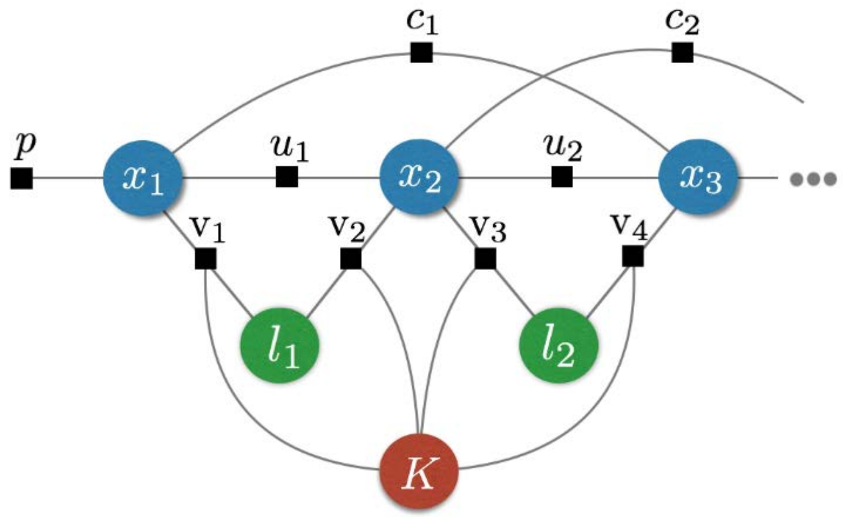
EFFICIENT MAP OPTIMIZATION AND SENSOR FUSION WITH FACTOR GRAPHS

Map optimization and sensor fusion with factor graphs

- Combining many different sensors in SLAM is a difficult and highly nonlinear problem

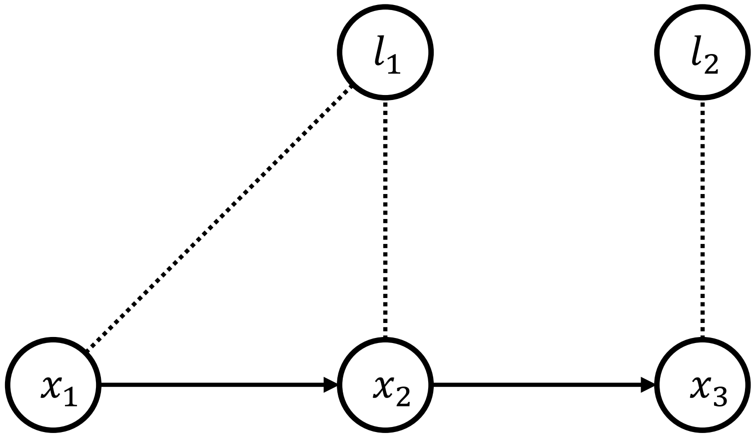
- Factor graphs provide powerful tools for expressing and solving nonlinear estimation problems

- It has become the current de-facto standard for the formulation of SLAM



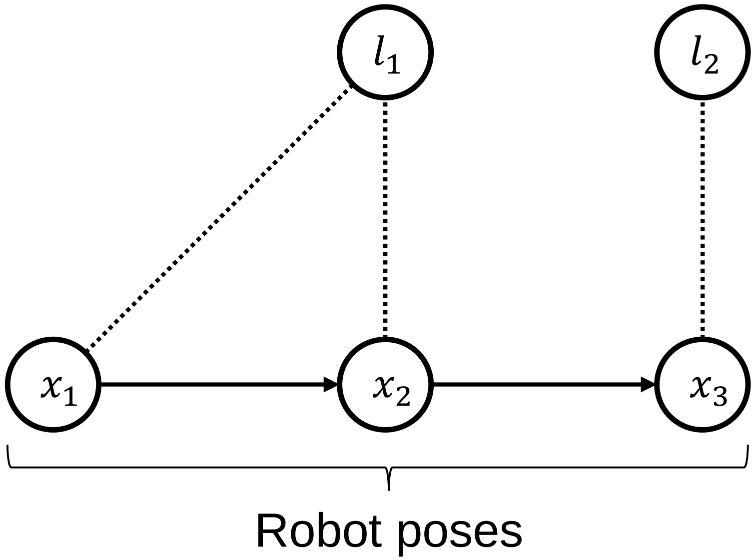
Cadena, C., et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332

Toy example



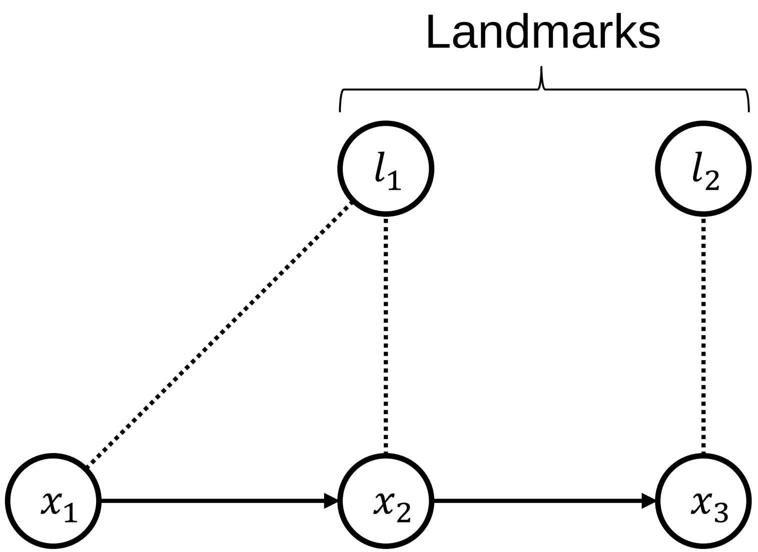
Toy example

Variables:



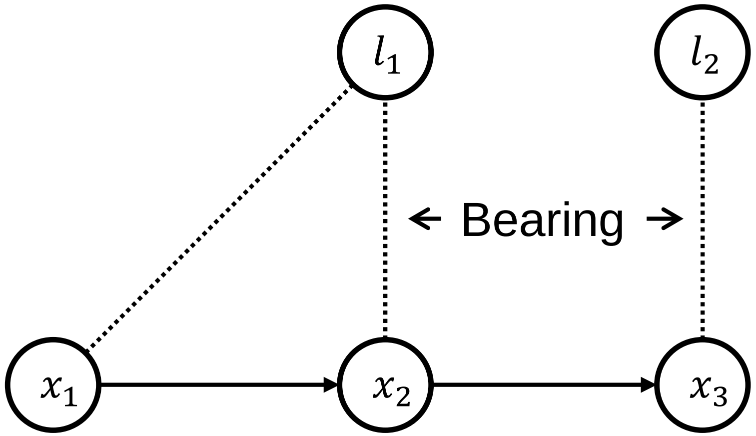
Toy example

Variables:



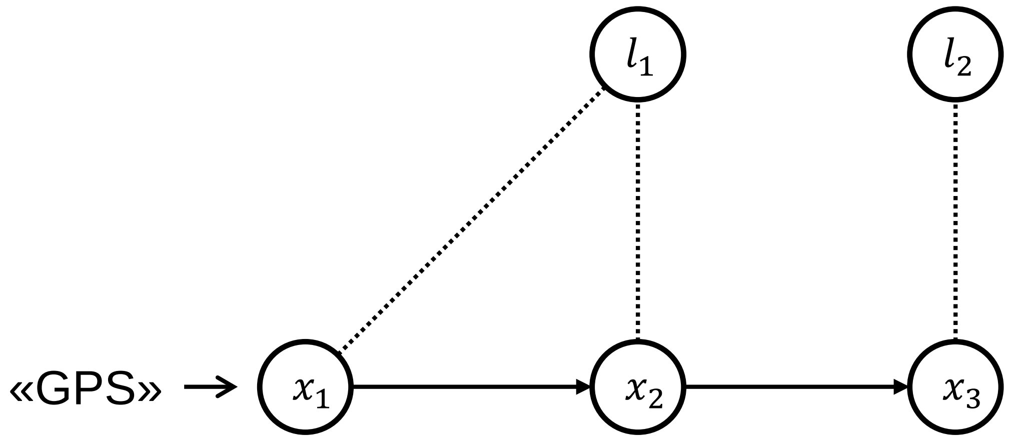
Toy example

Measurements:



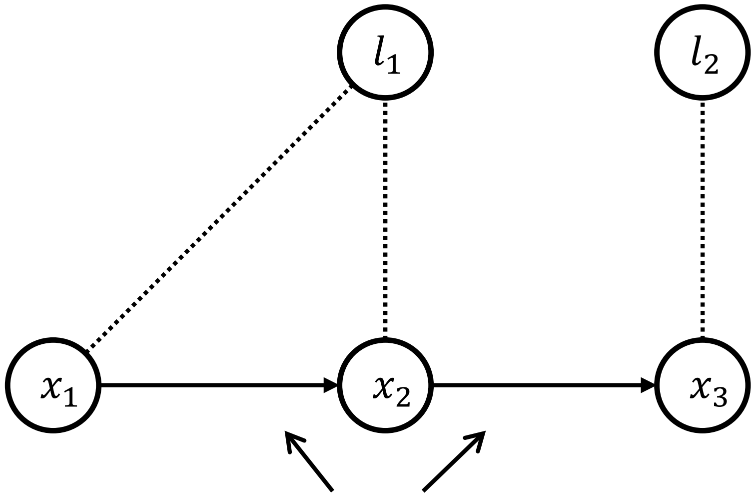
Toy example

Measurements:



Toy example

Motion model:



Toy example

Want to characterize our knowledge about the **unknown state variables**

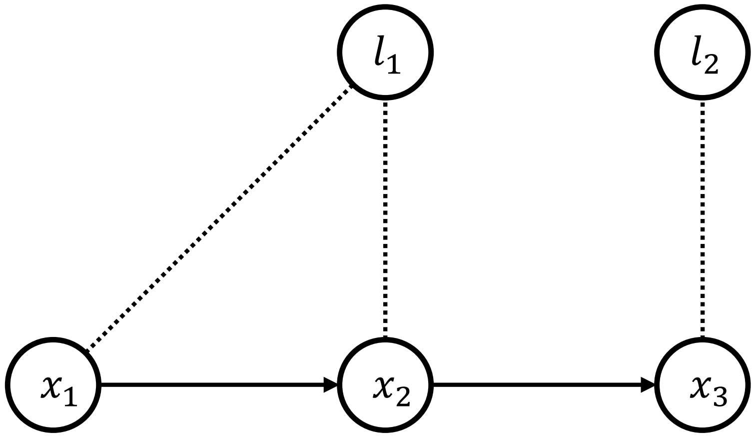
$$X = \{x_1, x_2, x_3, l_1, l_2\}$$

when *given* a set of **observed measurements**

$$Z = \{z_1, z_2, z_3, z_4\}$$

by obtaining

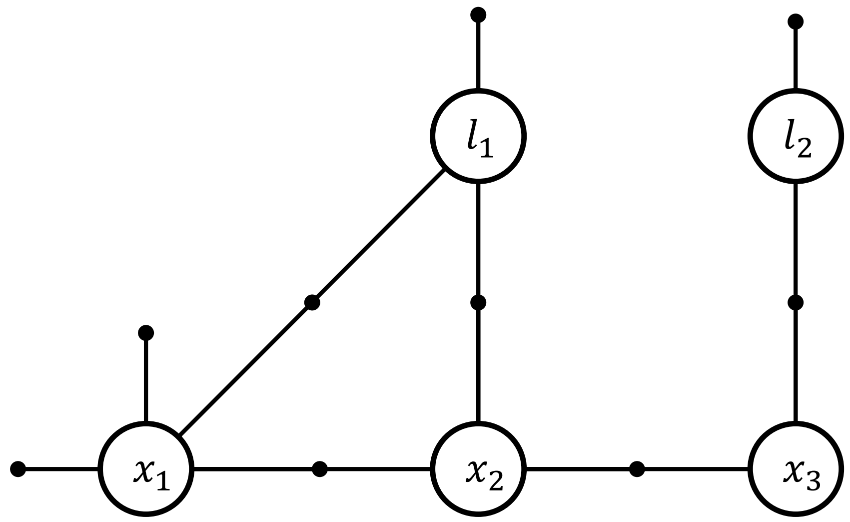
$$p(X \mid Z)$$



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

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

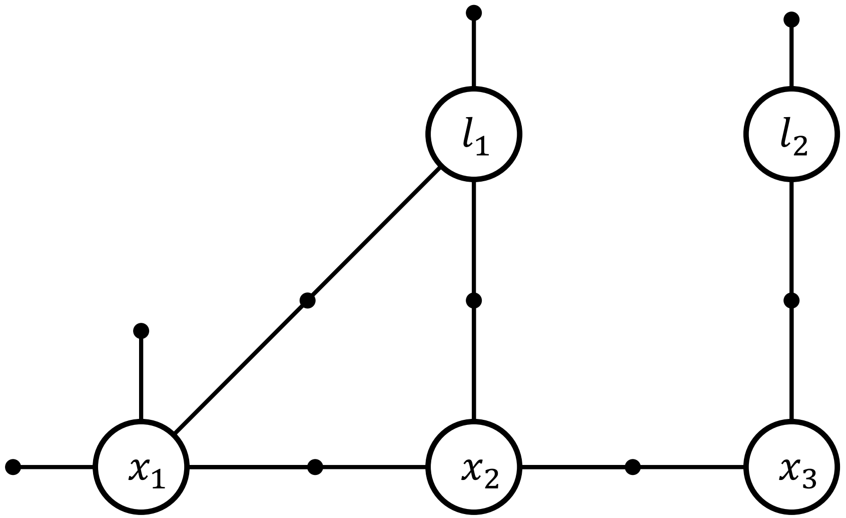
Taking the negative log and dropping the constant factor allows us instead to minimize a sum of *nonlinear least-squares*:

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

MAP inference for nonlinear factor graphs

MAP inference for factor graphs:

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



Let us assume that all factors are of the form

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

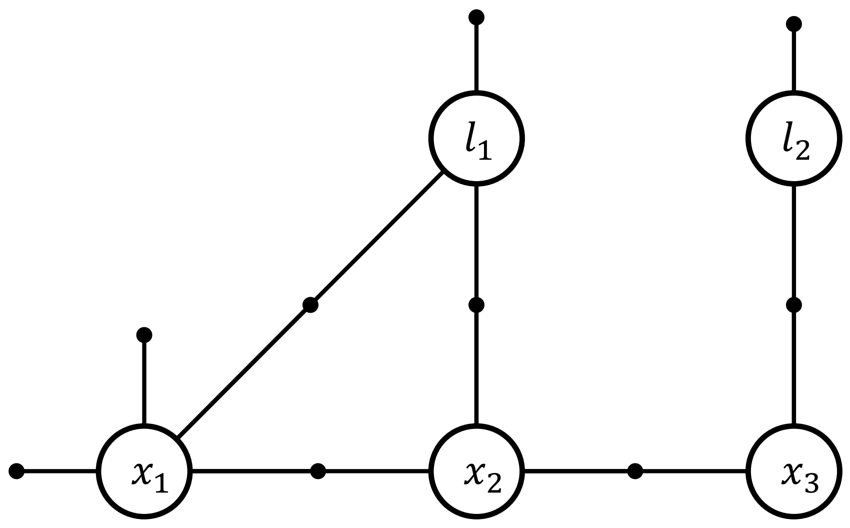
Taking the negative log and dropping the constant factor allows us instead to minimize a sum of *nonlinear least-squares*:

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

We know how to solve this!

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 $\mathbf{A}$.



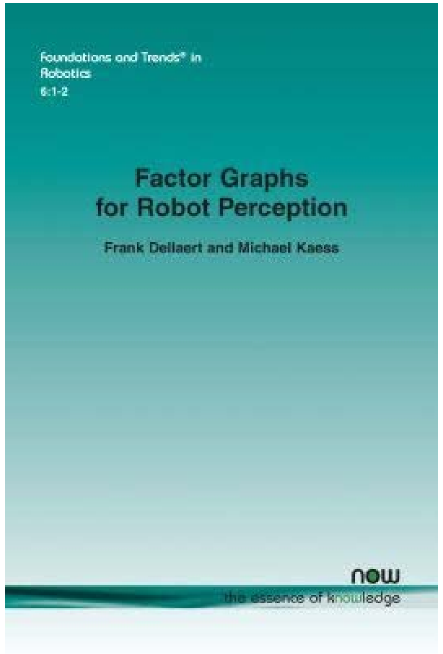
$$
\begin{array}{c} \\ \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{array}
\quad
[A \mid b] =
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & x_1 & x_2 & x_3 & b \end{array} \\
\left[\begin{array}{ccccc|c}
 & & A_{13} & & & b_1 \\
 & & A_{23} & A_{24}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

Supplementary material

- Georgia Tech Smoothing and Mapping library
  - https://bitbucket.org/gtborg/gtsam

- Jing Dong "GTSAM 4.0 Tutorial"

- Frank Dellaert "Factor Graphs and GTSAM: A Hands-on Introduction"
  Technical Report number GT-RIM-CP&R-2014-XXX
  September 2014
  (gtsam/doc/gtsam.pdf in the repo)



http://frc.ri.cmu.edu/~kaess/pub/Dellaert17fnt.pdf