

Sensors:

GNSS QR Code Trajectory

Camera Lidar IMU Odom

Applications:

UAV Self-driving Robots AR/VR

Bistatic Camera Model

State Estimation Problem

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k), & k = 1, \dots, K \\ \mathbf{z}_{k,j} = h(\mathbf{y}_j, \mathbf{x}_k, \mathbf{v}_{k,j}), & (k, j) \in \mathcal{O} \end{cases}$$

Stereo Camera

$$\begin{bmatrix} x_1 \\ x_2 \\ \theta \end{bmatrix}_k = \begin{bmatrix} x_1 \\ x_2 \\ \theta \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta \theta \end{bmatrix}_k + \mathbf{w}_k, \quad \begin{bmatrix} r_{k,j} \\ \phi_{k,j} \end{bmatrix} = \begin{bmatrix} \sqrt{(y_{1,j} - x_{1,k})^2 + (y_{2,j} - x_{2,k})^2} \\ \arctan\left(\frac{y_{2,j} - x_{2,k}}{y_{1,j} - x_{1,k}}\right) \end{bmatrix}$$

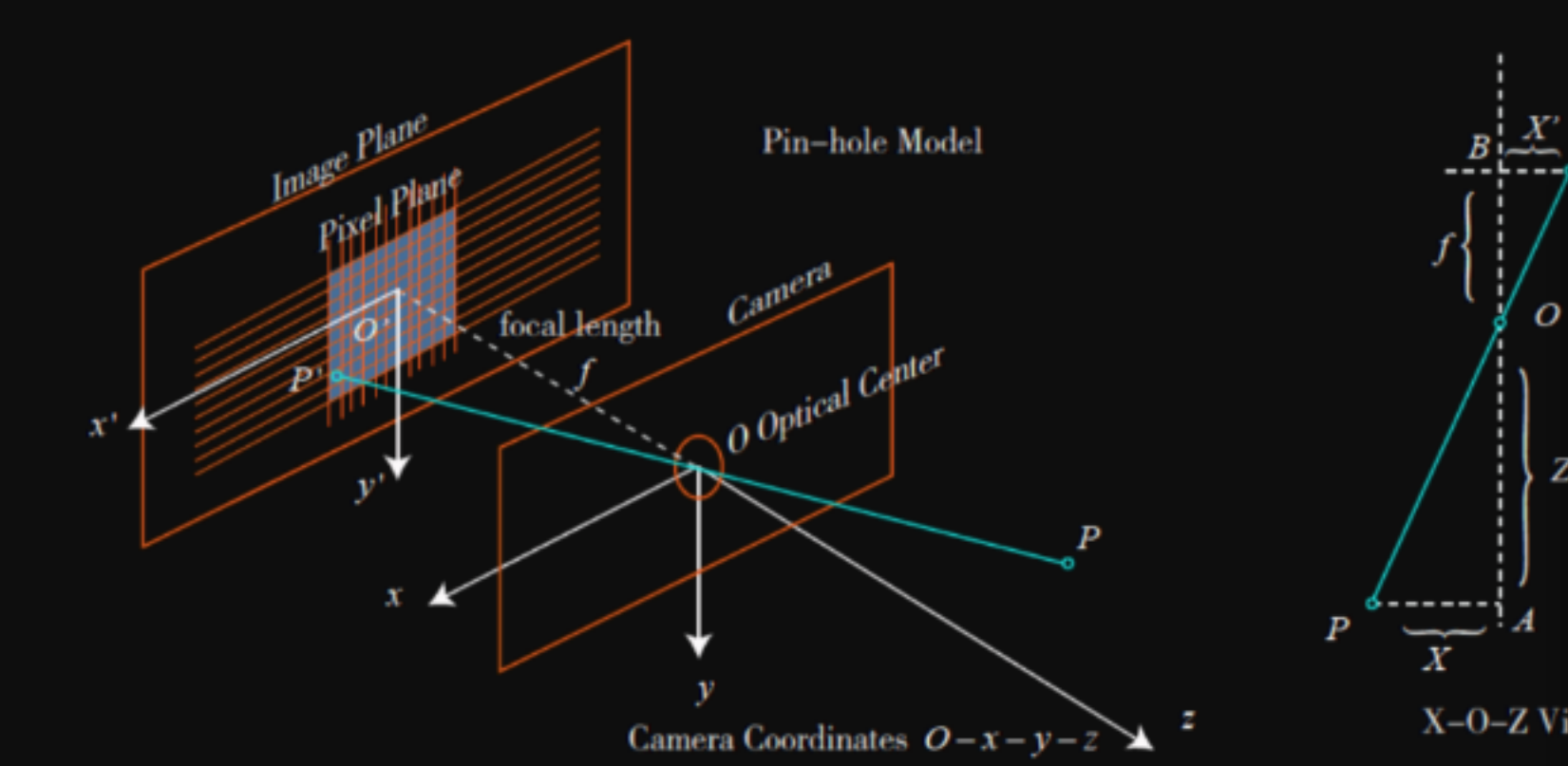
Euler Angles

Quaternion

Rodrigues' Formula

Baker-Campbell-Hausdorff Formula:

Left/Right Jacobians



Feature Matching

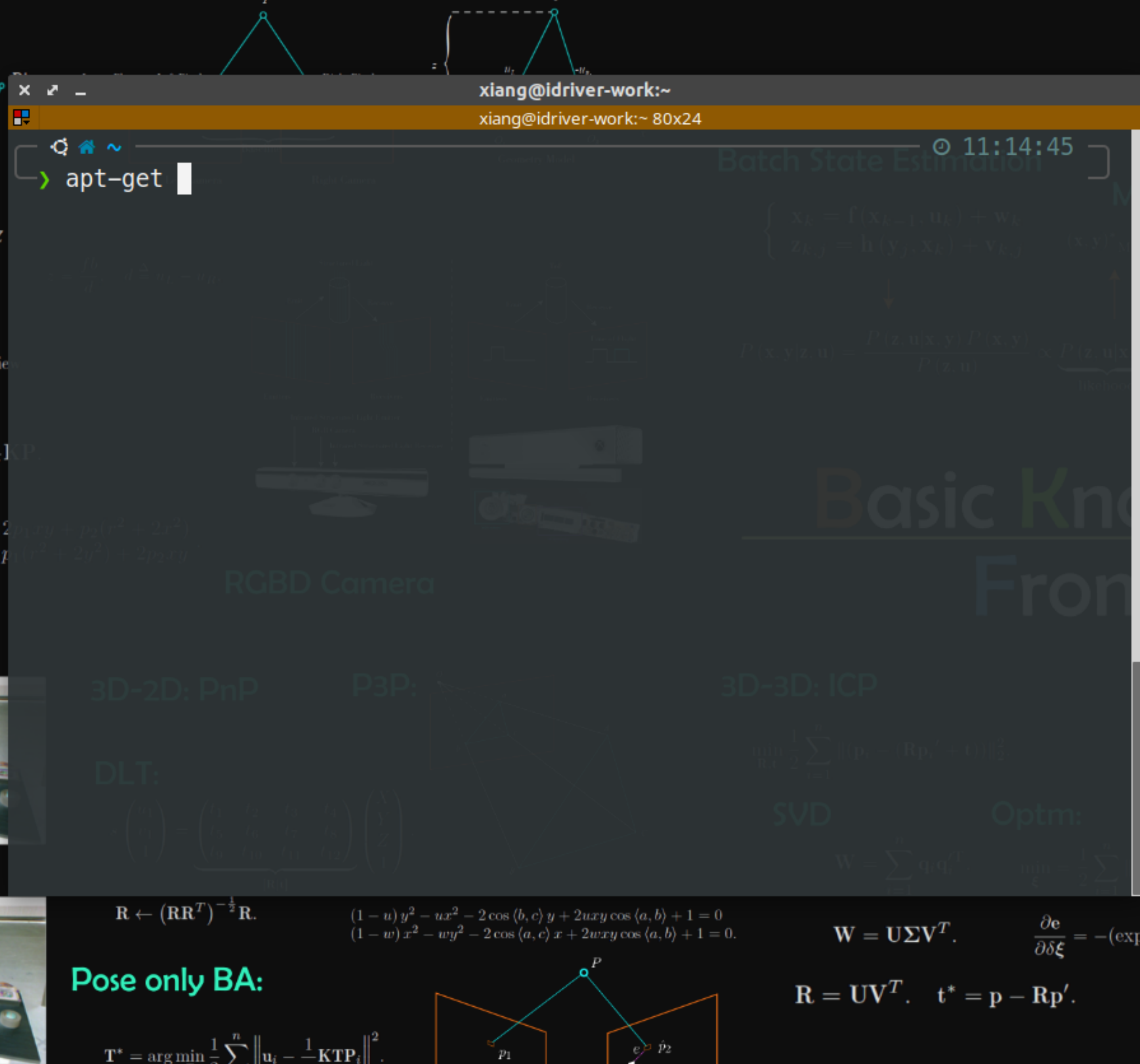
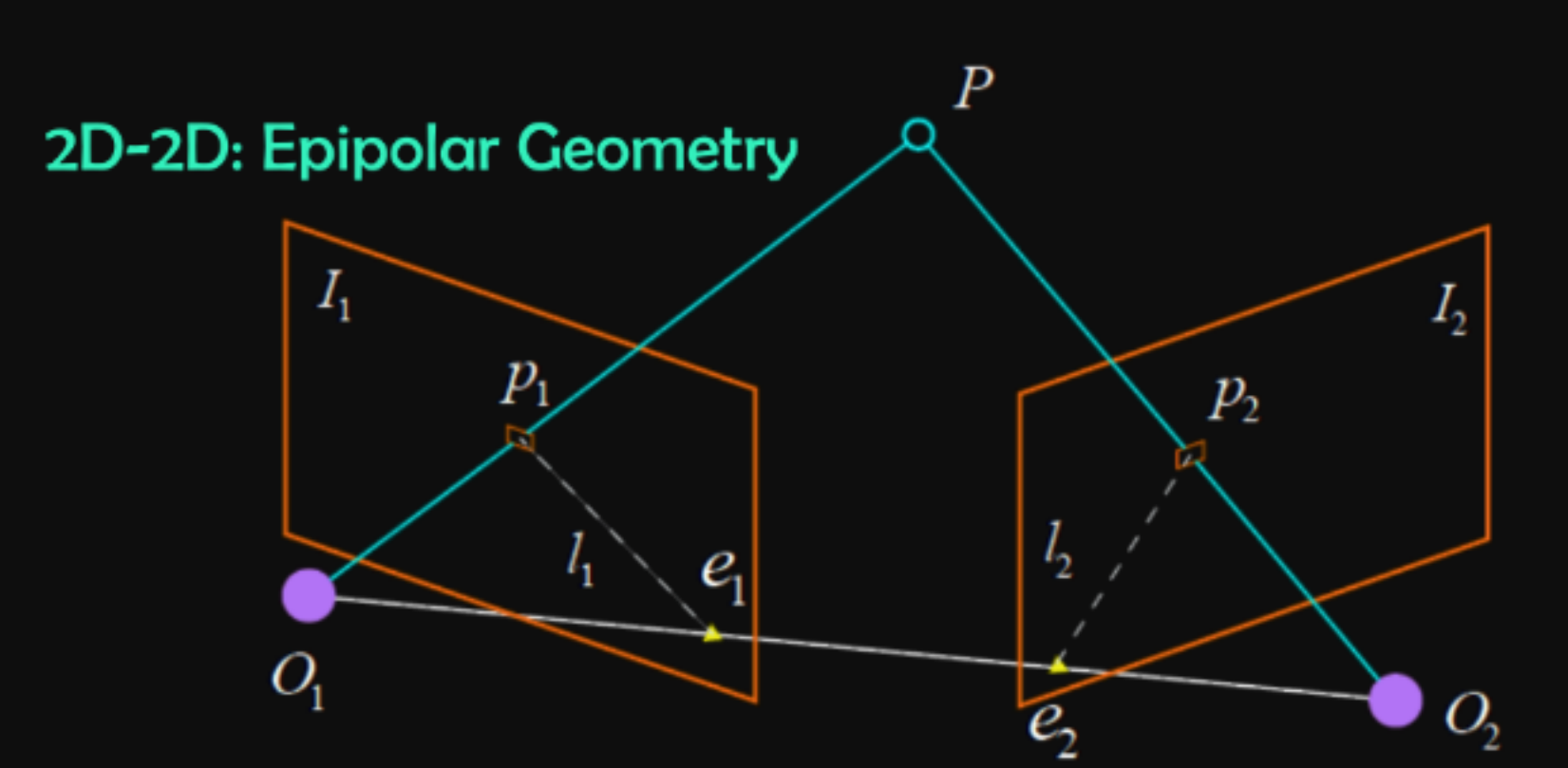
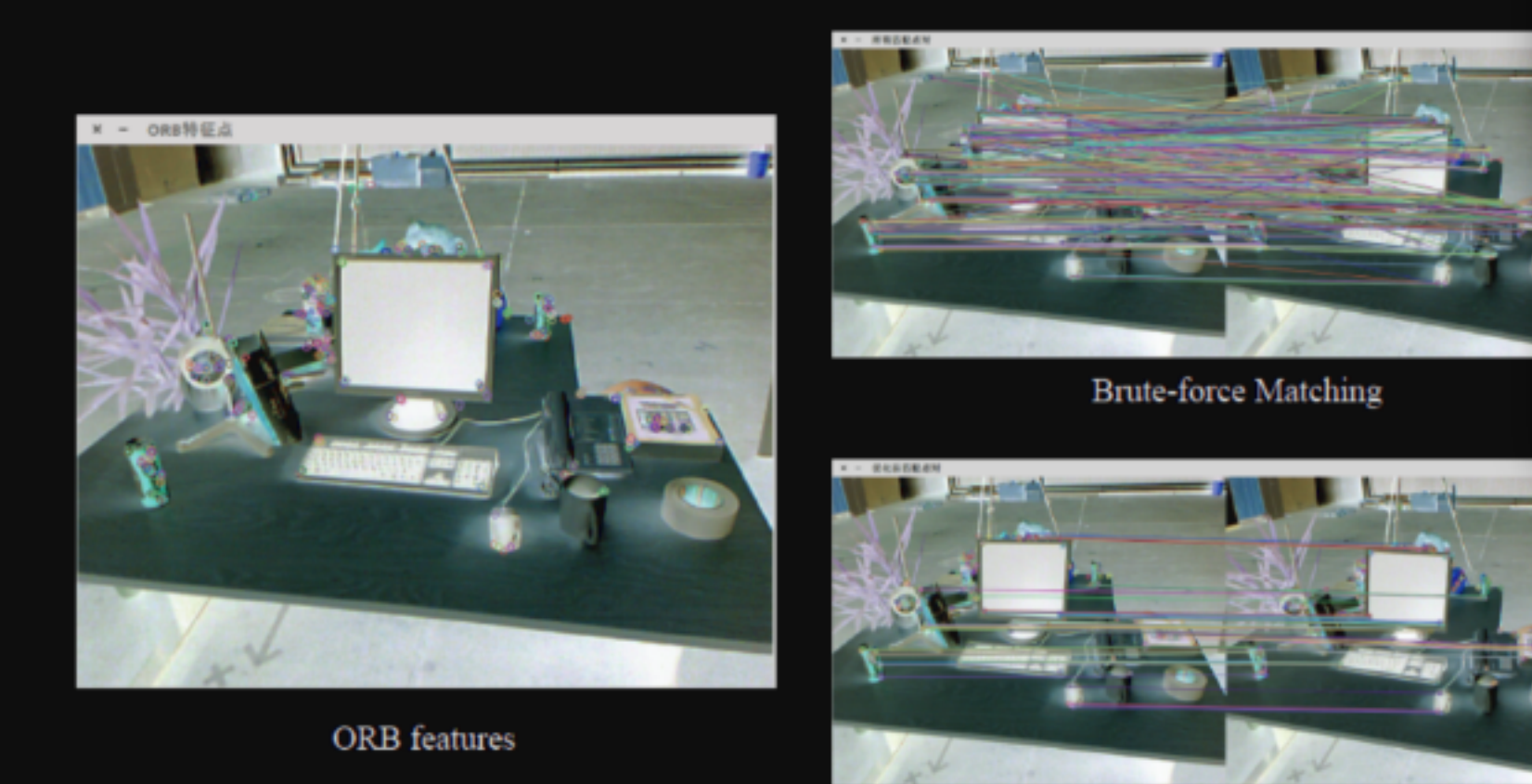
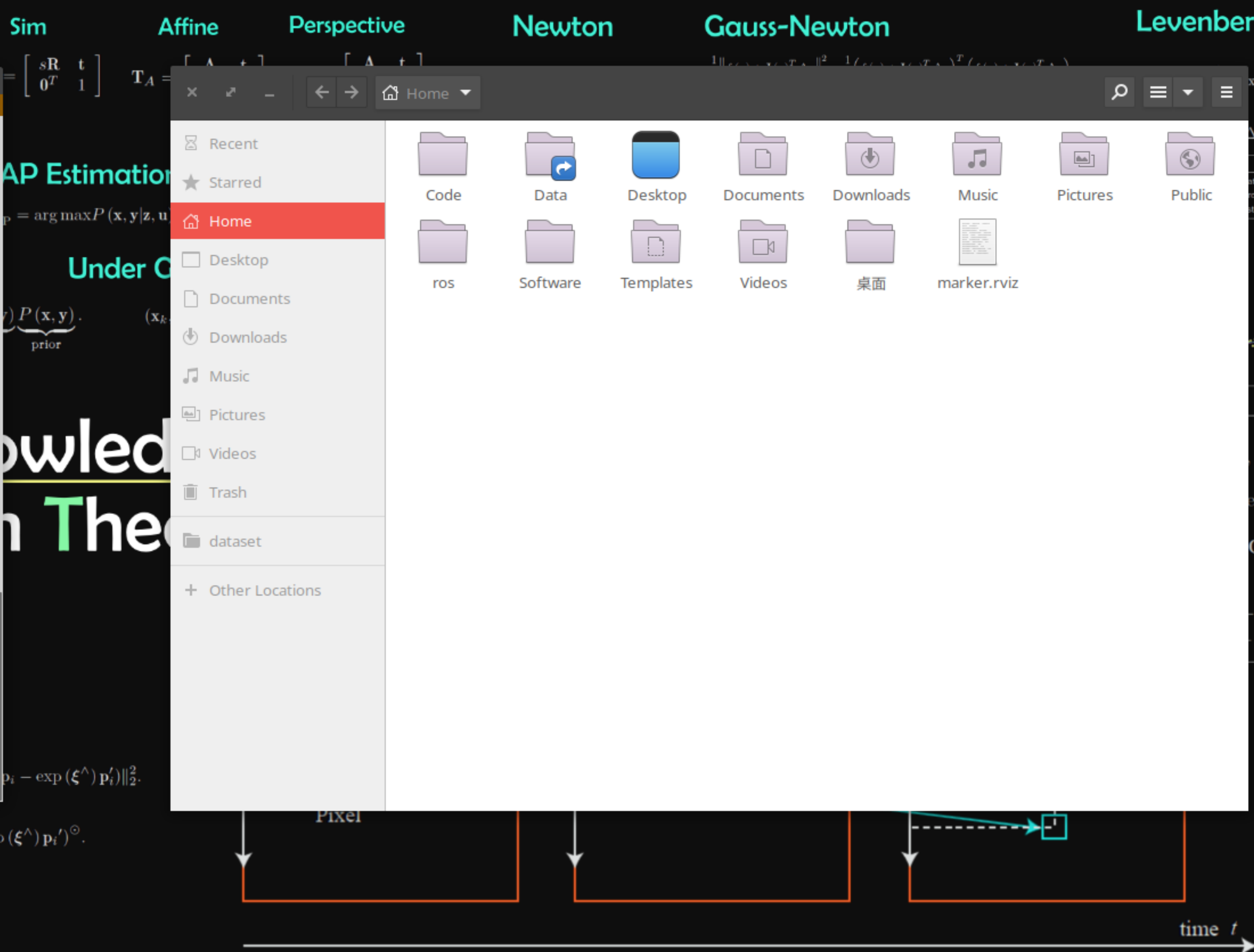


Figure 1: Triangulation. The figure illustrates the process of triangulation in a multi-camera system. The left part shows a 3D coordinate system with two cameras (C1, C2) and a point P. The point P is projected onto the image planes of both cameras, and the resulting epipolar lines are shown. The right part shows a multi-layer pyramid representing a coarse-to-fine tracking process. The bottom layer is labeled "Image 1" and the top layer is labeled "Raw image".

The Jacobian matrix for the projection function is given by:

$$\frac{\partial \mathbf{e}}{\partial \xi} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y X'}{Z'^2} & -\frac{f_y Y'}{Z'^2} & \frac{f_y X'}{Z'^2} & \frac{f_y Y'}{Z'} \end{bmatrix}$$



The diagram illustrates the Constant Grayscale Assumption and the Bundle Adjustment process. On the left, a 3D pyramid represents the scene, with a camera frustum labeled 'Image 2' at the bottom. The pyramid is divided into three horizontal layers, each associated with a time step t_1 , t_2 , and t_3 . The text 'Constant Grayscale Assumption' is written in green. Below it, the grayscale assumption is stated as $I(x + dx, y + dy, t + dt) = I(x, y, t)$. The bundle adjustment equations are shown as
$$\begin{bmatrix} \mathbf{I}_x & \mathbf{I}_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\mathbf{I}_t$$
 and
$$\mathbf{p}_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_i = \frac{1}{Z_i} \mathbf{K} \mathbf{p}_i$$
. The bundle adjustment process is shown as a sequence of images (a red apple, a folder, a green gear, a white camera, and a blue recycling symbol) with the text 'Bundle Adjustment' below them. On the right, a 'Pose Graph' is shown, consisting of a sequence of camera poses (triangles) connected by edges, with the text 'Pose Graph' above it. The pose graph equations are given as
$$e_{ij} = \Delta \xi_{ij} \ln (T_{ij}^{-1} T_i^{-1} T_j)^\vee$$
 and
$$e_{ij} = \ln (T_{ij}^{-1} T_i^{-1} \exp((- \delta \xi_i)^\wedge) \exp(\delta \xi_j^\wedge) T_j)$$
.

g-Marquardt

$$\mathbf{J}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k)^T \Delta \mathbf{x}_k \Big\|^2 + \frac{\lambda}{2} \left(\|\mathbf{D} \Delta \mathbf{x}_k\|^2 - \mu \right),$$

$$\Delta \mathbf{x}_k = \mathbf{g}_k.$$

Trust region

Graph Optimization

○ Landmarks
 △ Camera poses
 - - - Observation model
 — Motion model

$$\hat{\mathbf{P}}_k = \mathbf{A}_k \hat{\mathbf{P}}_{k-1} \mathbf{A}_k^T + \mathbf{R}_k. \quad (8.24)$$

Kalman gain:

$$\mathbf{C}_k \hat{\mathbf{P}}_k \mathbf{C}_k^T + \mathbf{Q}_k)^{-1}, \quad (8.25)$$

$$\mathbf{K}(\mathbf{z}_k - \mathbf{C}_k \mathbf{x}_k) \quad (8.26)$$

$$\mathbf{K} \mathbf{C}_k^T \hat{\mathbf{P}}_k.$$

Sparse BA

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}.$$

$$\mathbf{J} = \begin{bmatrix} \square & \square & \square & \square & \square & \square & \square \\ \uparrow & & & & & & \\ & & \square & & & & \\ & & & \square & & & \\ & & & & \square & & \\ & & & & & \square & \\ & & & & & & \square \end{bmatrix} \quad \mathbf{H} += \begin{bmatrix} & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{P}_1 & \mathbf{P}_2 & \mathbf{P}_3 & \mathbf{P}_4 & \mathbf{P}_5 \\ \mathbf{C}_1^T & \mathbf{C}_2^T & \mathbf{P}_1^T & \mathbf{P}_2^T & \mathbf{P}_3^T & \mathbf{P}_4^T & \mathbf{P}_5^T \end{bmatrix}$$

Arrow-like Hessian

Schur-trick

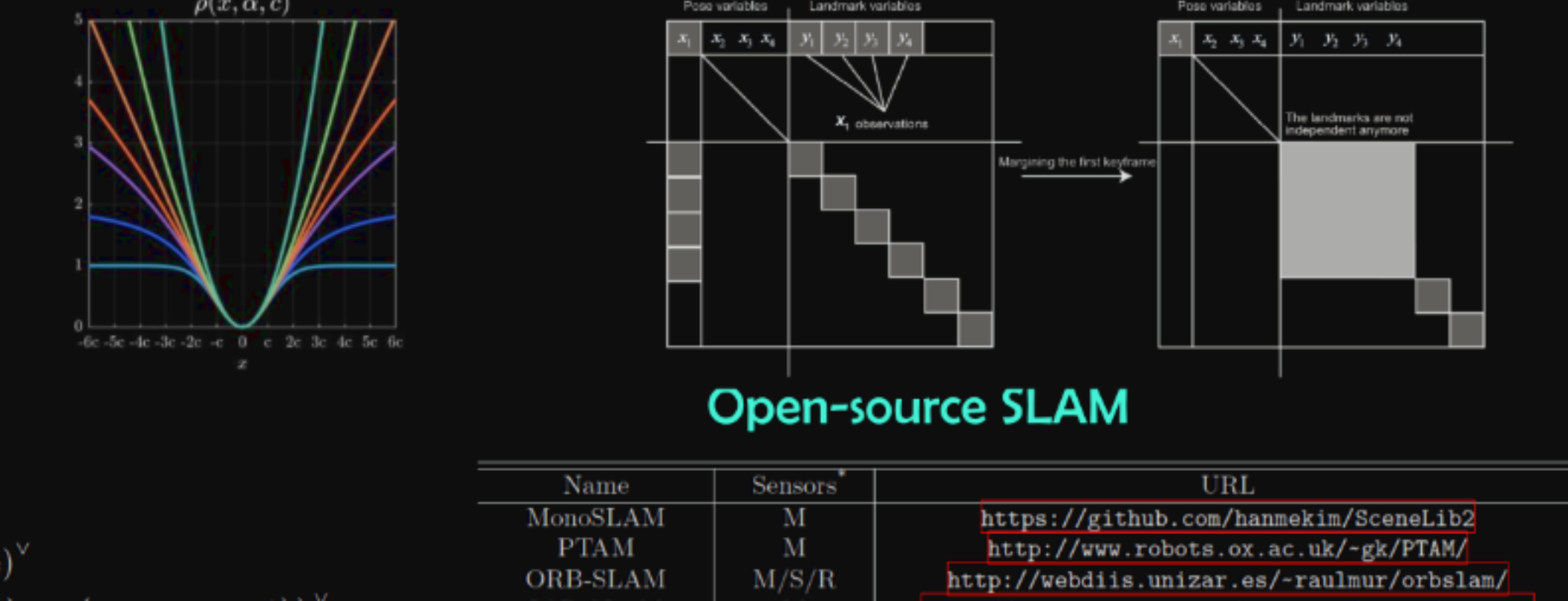
$$\begin{bmatrix} \mathbf{I} & -\mathbf{E} \mathbf{C}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_c \\ \Delta \mathbf{x}_p \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{E} \mathbf{C}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}.$$

$$\begin{bmatrix} \mathbf{B} - \mathbf{E} \mathbf{C}^{-1} \mathbf{E}^T & \mathbf{0} \\ \mathbf{E}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_c \\ \Delta \mathbf{x}_p \end{bmatrix} = \begin{bmatrix} \mathbf{v} - \mathbf{E} \mathbf{C}^{-1} \mathbf{w} \\ \mathbf{w} \end{bmatrix}.$$

$$[\mathbf{B} - \mathbf{E} \mathbf{C}^{-1} \mathbf{E}^T] \Delta \mathbf{x}_c = \mathbf{v} - \mathbf{E} \mathbf{C}^{-1} \mathbf{w}.$$

Robust kernel

Fill-in in Marginalization



Name	Sensors*	URL
MonoSLAM	M	https://github.com/hanmekim/SceneLib2
PTAM	M	http://www.robots.ox.ac.uk/~gk/PTAM/
ORB-SLAM	M/S/R	http://webdiis.unizar.es/~raulmur/orbslam/