The problem

- How do we accurately detect ego-motion using optical flow?

- What can we do now that we know how the camera has moved?

Current Objectives

- Accurately detect how much the system has moved
- Accurately detect how much the system has rotated
The experiment

- What’s the optical flow in this sequence?
The literature

- Read and comprehended the following papers:
  - Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm – Bouguet
  - Passive Navigation – Bruss and Horn
  - Ego-motion Estimation Using Optical Flow Fields Observed from Multiple Cameras – Tsao et al
The setup

\[ C_g = \{ O, I = [e_1 | e_2 | e_3] \} \]

\[ C_k = \{ b_k, R_k = [u_{k1} | u_{k2} | u_{k3}] \} \]
The problem

- Relative to Cg, the following is true from the perspective projection model:

\[ \hat{P}_k = -\omega_k \times P_k - t_k \]

\[ p_k \equiv \begin{bmatrix} p_{xk} \\ p_{y_k} \\ 1 \end{bmatrix} = \frac{1}{P_{zk}} P_k \]

- Differentiate (2) and substitute (1):

\[ v_k \equiv \dot{p}_k = - (\omega_k \times p_k) - \frac{\dot{P}_{zk}}{P_{zk}} P_k - \frac{1}{P_{zk}} t_k \]
The problem

- Plug-in the following and simplify:
  \[ \omega_k \equiv R_k^T \omega \quad \text{and} \quad t_k \equiv R_k^T [\omega \times b_k] + t \]

- To get:
  \[ m_{ki}^T (h_k + t) = 0 \]
  \[ h_k \equiv \omega \times b_k \]
  \[ m_{ki} \equiv R_k \{ p_{ki} \times [\dot{p}_{ki} + (R_k^T \omega \times p_{ki})] \} \]
The problem

- Take this:
  \[ J'_1(\omega, t) \equiv \sum_{k=1}^{K} \sum_{i=1}^{N_k} \| m_{ki}^T (h_k + t) \|^2 \]

- And solve for t:
  \[ t = M^{-1} c \]

- Plug t back in, now minimize the following:
  \[ J_1(\omega) \equiv -c^T M^{-1} c + \sum_{k=1}^{K} \sum_{i=1}^{N_k} (m_{ki}^T h_k)^2 \]
The experiment
The problem

3D grid approach