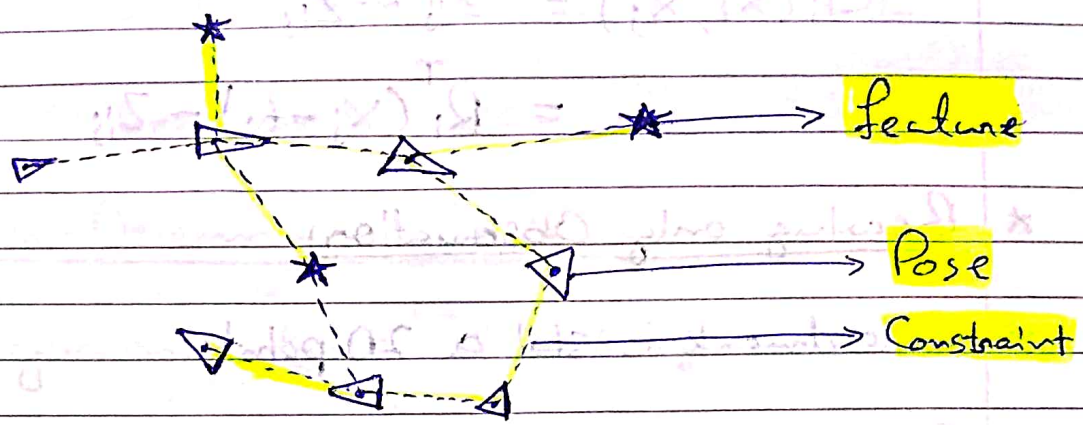
(6)

Graph based SLAM with Landmarks

★ The Graph with Landmarks



⇒ Nodes can represent:
- → Robot poses
- → Landmark location

⇒ Edges can represent:
- → Landmark observations
- → Odometry measurements.

⇒ The minimization optimizes the landmark locations and robot poses.

★ Landmarks observation

⇒ Expected observation

$$\hat{z}_{ij}(x_i, x_j) = R_i^T (x_j - t_i)$$

x_i: Robot, x_j: landmark, t_i: Robot translation

⇒ Error function

$$e_{ij}(x_i, x_j) = \hat{z}_{ij} - z_{ij}$$

$$= R_i^T (x_j - t_i) - z_{ij}$$

* Bearing only Observation

⇒ A landmark is still a 2D point.

⇒ The robot observe only the bearing towards the landmark.

⇒ Observation function

$$\hat{z}_{ij}(x_i, x_j) = \text{atan}\left(\frac{(x_j - t_i) \cdot y}{(x_j - t_i) \cdot x}\right) - \theta_i$$

⇒ Error function

$$e_{ij}(x_i, x_j) = \text{atan}\left(\frac{(x_j - t_i) \cdot y}{(x_j - t_i) \cdot x}\right) - \theta_i - z_j$$

* The Rank of the Matrix H

① Rank of H_{ij} for 2D landmark pose Constraint

- The block of J_{ij} are at most 2×5 matrix
- H_{ij} Cannot have more than rank 2

② Rank of H_{ij} for a bearing-only constraint:

→ The block of J_{ij} are at most 1×5 matrix

→ H_{ij} has rank 1.

★ Under-Determined Systems

⇒ No guarantee for a full rank system.

⇒ We can still deal with these situations by adding a "damping" factor to H.

⇒ Instead of solving $H\Delta x = -b$

we solve,

$$(H + \lambda I)\Delta x = -b$$

★ Simplified Levenberg Marquardt

```
1.  X: the initial guess
2.  while (! Converged)
3.      λ = λinit;
4.      <H, b> = buildLinearSystem(x);
5.      E = error(x);
6.      Xold = X;
7.      Δx = SolveSparse((H + λI)Δx = -b);
8.      x = x + Δx;
9.      If (E < error(x)) {
10.         x = Xold;
11.         λ *= 2;
12.     } else { λ /= 2; }
```

⇒ Damping to negate the convergence using backup/restore actions.

* Bundle Adjustment

↳ 3D reconstruction based on images taken at different viewpoint.