Robot Mapping

Assignment: Sheet 10 ,Graph - Based SLAM

Authors:

Elias M Maharmeh

1 Exercise 1: Global Error

- Local error vector between two nodes (i, j), $\mathbf{e}_{ij} = t2v(Z_{ij}^{-1}(X_i^{-1}X_j))$
- Local error vector between a node and a landmark (i, l), $\mathbf{e}_{il} = t2v(Z_{il}^{-1}(X_i^{-1}X_l))$
- Global error is the sum of the weighted magnitude of the local errors

$$\mathbf{F}(\mathbf{x}) = \sum_{(i,j)\in\mathcal{C}} \mathbf{F_{ij}}(\mathbf{x}) = \sum_{(i,j)\in\mathcal{C}} e_{ij}^T \Omega_{ij} e_{ij} \tag{1}$$

  where j could be a node pointer or landmark pointer.

- The **Goal** is to find the maximum likelihood spatial configuration that best fit and explain the measurements, i.e minimize the global error function $\mathbf{F}(\mathbf{x})$

2 Exercise 2: linearization process

We need to compute the Jacobian's non-zero elements by taking the derivative of the error function about the current $\hat{x}$ estimate, and perform it at each iteration step to updtae the jacobian. The following pics illustrate the derivation process.

$$e_{ij} = \begin{pmatrix} R_{ij}^T\left(R_i^T(x_j - x_i) - t_{ij}\right) \\ \theta_i - \theta_i - \theta_{ij} \end{pmatrix} = \begin{pmatrix} e_{ij}(1) \\ e_{ij}(2) \end{pmatrix}$$

Figure 1: error definition

$$\frac{\partial e_{ij}(1)}{\partial x_i} = R_{ij}^T \cdot \frac{\partial}{\partial x_i} R_i^T(x_j - x_i)$$

Figure 2: derivative of error pose w.r.t x_i

$$x_i = (x_i, y_i, \theta_i)^T$$
$$x_j = (x_j, y_j, \theta_j)^T$$

$$R_i^T = \begin{bmatrix} c\theta_i & s\theta_i \\ -s\theta_i & c\theta_i \end{bmatrix}$$

Figure 3: definition of the pose vector and rotation matrix

$$\frac{\partial}{\partial x_i}\left(\begin{bmatrix} c\theta_i & s\theta_i \\ -s\theta_i & c\theta_i \end{bmatrix}\begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix}\right)$$

Figure 4: derivative of the rotated position error of the two nodes

$$= \left[\begin{array}{c|c|c} -c\theta_i & -s\theta_i & -(x_j - x_i)s\theta_i + (y_j - y_i)c\theta_i \\ s\theta_i & -c\theta_i & -(x_j - x_i)c\theta_i - (y_j - y_i)s\theta_i \end{array}\right]$$

$$= \left[\begin{array}{c|c} -R_i^T & \begin{matrix} -dx_{ij}\, s\theta_i + dy_{ij}\, c\theta_i \\ -dx_{ij}\, c\theta_i - dy_{ij}\, s\theta_i \end{matrix} \end{array}\right]$$

Figure 5: derivative of the rotated position error of the two nodes

$$\therefore \frac{\partial e_{ij(2)}}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\theta_j - \theta_i - \theta_{ij}\right) = \left(0 \;\; 0 \;\; -1\right)$$

Figure 6: derivative of the orientation error

$$A_{ij} = \frac{\partial e_{ij}}{\partial x_i} = \left(\begin{array}{cc|c} \multicolumn{2}{c|}{-R_i^T} & \begin{matrix} -dx_{ij}\, s\theta_i + dy_{ij}\, c\theta_i \\ -dx_{ij}\, c\theta_i - dy_{ij}\, s\theta_i \end{matrix} \\ \hline 0 & 0 & -1 \end{array}\right)_{3\times 3}$$

Figure 7: derivative of the error vector w.r.t node x_i

The same procedure is applied to the pose-landmark error linearization.

3 Exercise 3 : Gauss-Newton procedure

The following algorithm[1] summarizes an iterative Gauss-Newton procedure to determine both the mean and the information matrix of the posterior over the robot poses

[1] A Tutorial on Graph-Based SLAM Giorgio Grisetti Rainer K¨ummerle Cyrill Stachniss Wolfram Burgard Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany

Algorithm 1 Computes the mean $\mathbf{x}^*$ and the information matrix $\mathbf{H}*$ of the multivariate Gaussian approximation of the robot pose posterior from a graph of constraints.

Require: $\breve{\mathbf{x}} = \breve{\mathbf{x}}_{1:T}$: initial guess. $\mathcal{C} = \{\langle \mathbf{e}_{ij}(\cdot), \mathbf{\Omega}_{ij} \rangle\}$: constraints
Ensure: $\mathbf{x}^*$: new solution, $\mathbf{H}^*$ new information matrix
// find the maximum likelihood solution
while $\neg$converged **do**
$\mathbf{b} \leftarrow \mathbf{0}$ $\quad$ $\mathbf{H} \leftarrow \mathbf{0}$
for all $\langle \mathbf{e}_{ij}, \mathbf{\Omega}_{ij} \rangle \in \mathcal{C}$ **do**
// Compute the Jacobians $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ of the error function
$\mathbf{A}_{ij} \leftarrow \left.\frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_i}\right|_{\mathbf{x}=\breve{\mathbf{x}}}$ $\quad$ $\mathbf{B}_{ij} \leftarrow \left.\frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_j}\right|_{\mathbf{x}=\breve{\mathbf{x}}}$
// compute the contribution of this constraint to the linear system
$\mathbf{H}_{[ii]} \mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij}$ $\quad$ $\mathbf{H}_{[ij]} \mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij}$
$\mathbf{H}_{[ji]} \mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij}$ $\quad$ $\mathbf{H}_{[jj]} \mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij}$
// compute the coefficient vector
$\mathbf{b}_{[i]} \mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}$ $\quad$ $\mathbf{b}_{[j]} \mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}$
end for
// keep the first node fixed
$\mathbf{H}_{[11]} \mathrel{+}= \mathbf{I}$
// solve the linear system using sparse Cholesky factorization
$\mathbf{\Delta x} \leftarrow \text{solve}(\mathbf{H}\,\mathbf{\Delta x} = -\mathbf{b})$
// update the parameters
$\breve{\mathbf{x}} \mathrel{+}= \mathbf{\Delta x}$
end while
$\mathbf{x}^* \leftarrow \breve{\mathbf{x}}$
$\mathbf{H}^* \leftarrow \mathbf{H}$
// release the first node
$\mathbf{H}^*_{[11]} \mathrel{-}= \mathbf{I}$
return $\langle \mathbf{x}^*, \mathbf{H}^* \rangle$