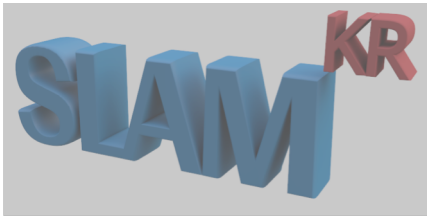
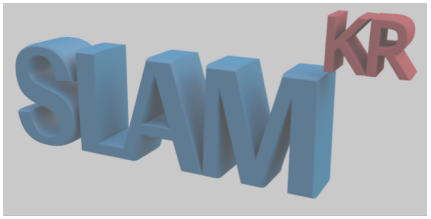


Backend 2

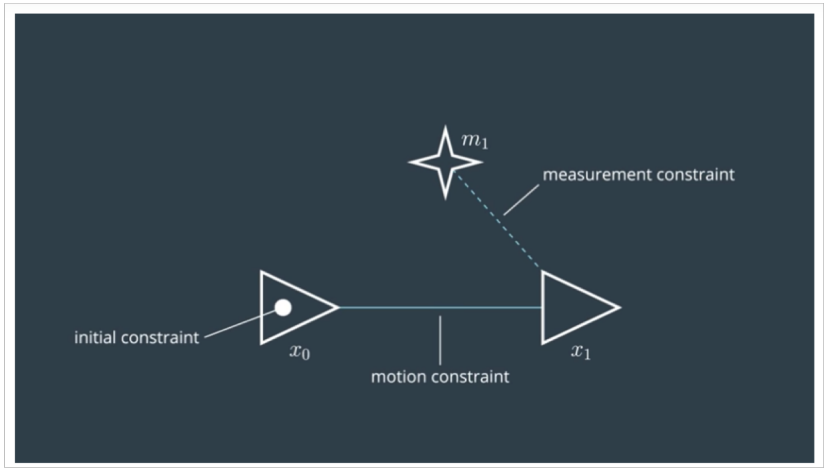
March 24, 2019

Dong-Won Shin

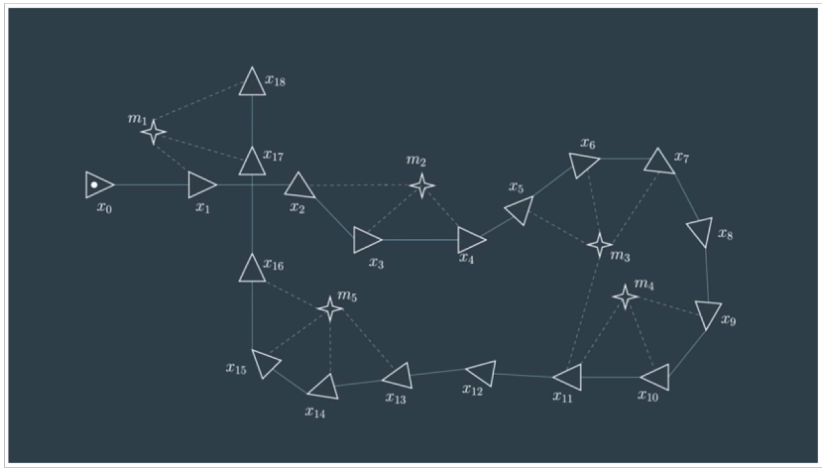
GraphSLAM



- Graph-based formulation of the SLAM problem by Lu and Milios in 1997
- Build the graph and find the robot poses and landmarks that minimize the error

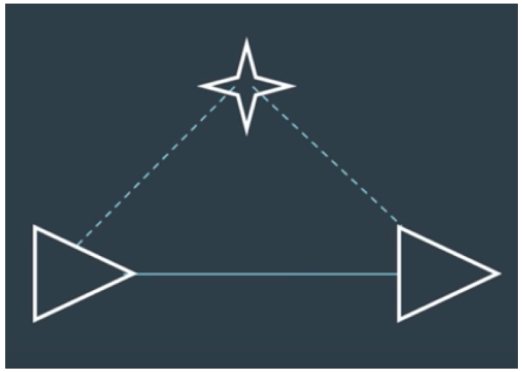


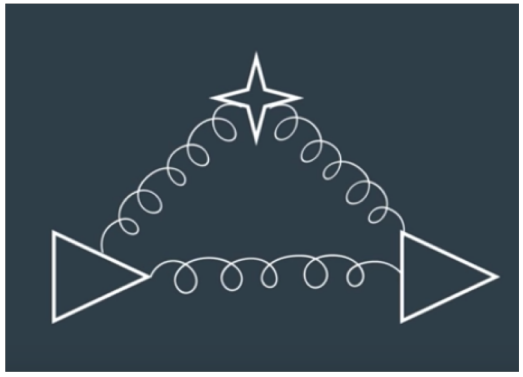
Simple example



Complicated example

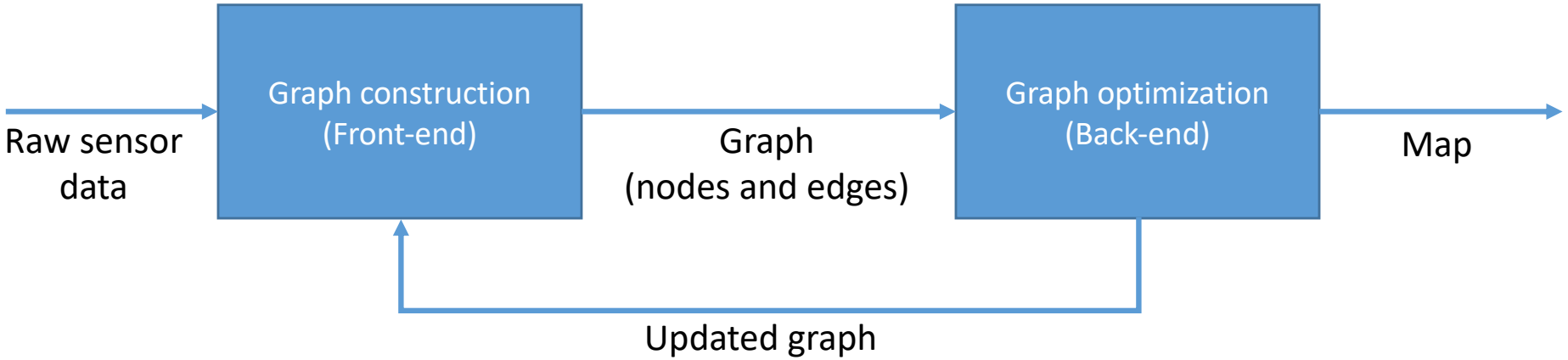
- Spring-mass model





E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), pages 2262–2269, 2006.

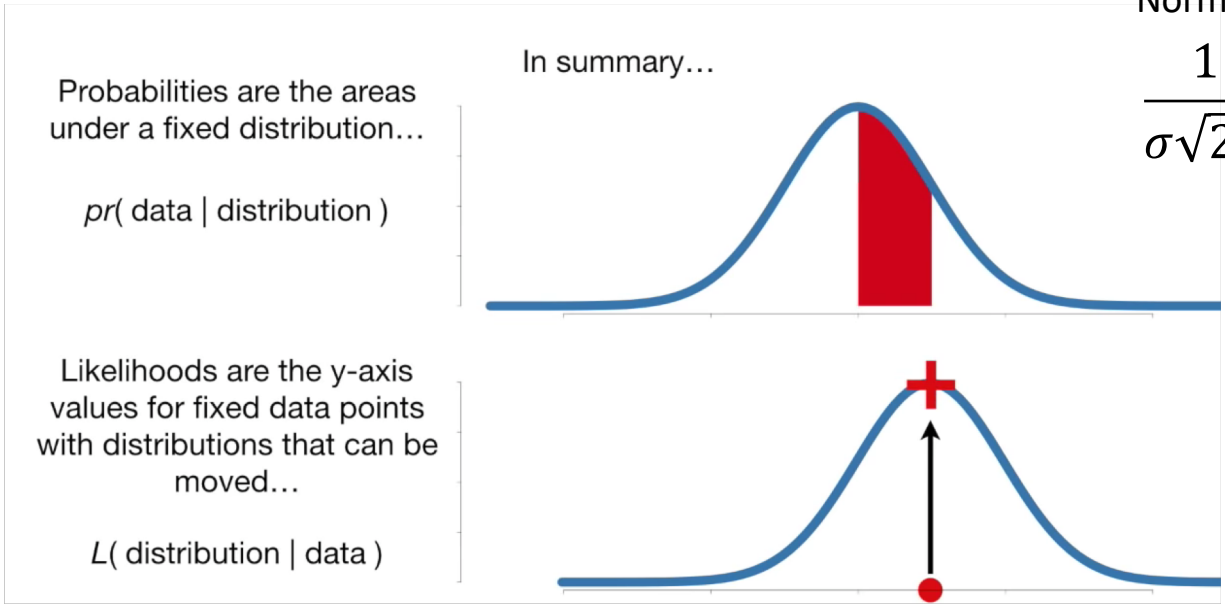
Front-end and Back-end



- Front-end
  - How to construct the graph using the odometry and sensory measurements by the robot
  - Data association
- Back-end
  - Input
    - The completed graph with all of the constraints
  - Output
    - The most probable configuration of robot poses and map features
  - Graph optimization
    - Find the system configuration that produces the smallest error

Maximum Likelihood Estimation

- Probability
  - 주어진 확률분포에서 해당 관측값이 나올 비율
- Likelihood
  - 확률과는 상보적인 개념
  - 주어진 관측값에서 이것이 해당 확률 분포에서 나올 비율

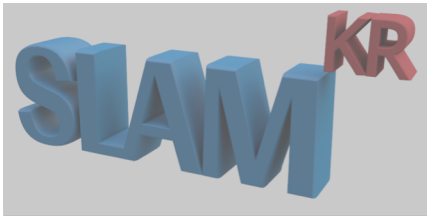


Normal distribution

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

https://jjangjjong.tistory.com/41
https://youtu.be/pYxNSUDSFH4

1-Dimensional Case



$$\arg\max_{x,z} p(\mathbf{x}_{1:T}, \mathbf{m}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T}) = \prod p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$$

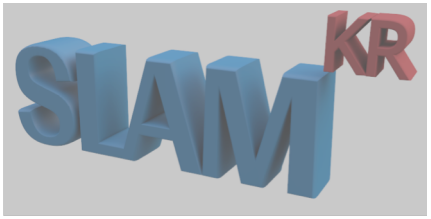
Measurement constraint

Motion constraint

- Example of 1-Dimensional graph
  - 1-D Measurement constraint: $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}_t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\mathbf{z}_t-(\mathbf{x}_t+\mathbf{m}_t))^2}{\sigma^2}\right\}$
  - 1-D Motion constraint: $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\mathbf{x}_t-(\mathbf{x}_{t-1}+\mathbf{u}_t))^2}{\sigma^2}\right\}$

- Remove scaling factors
  - Parameters that maximizes the equation does not depend on the constraints in front of each of the exponentials
  - 1-D Measurement constraint: $\exp\left\{-\frac{(z_t-(x_t+m_t))^2}{\sigma^2}\right\}$
  - 1-D Motion constraint: $\exp\left\{-\frac{(x_t-(x_{t-1}+u_t))^2}{\sigma^2}\right\}$

1-Dimensional Case

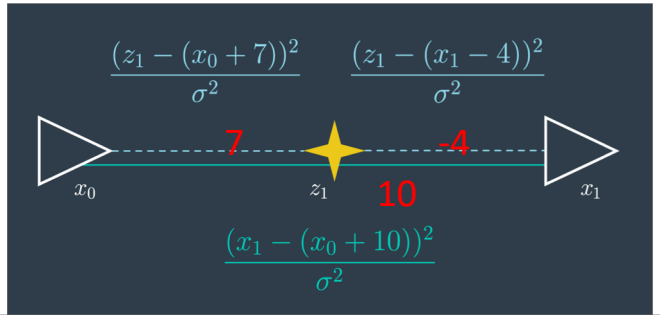


$$\arg\max_{x,z} p(\mathbf{x}_{1:T}, \mathbf{m}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T}) = \prod p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$$

Measurement constraint

Motion constraint

- Negative-Log-Likelihood
  - $-\prod \log\left(\exp\left\{-\frac{(z_t-(x_t+m_t))^2}{\sigma^2}\right\}\exp\left\{-\frac{(x_t-(x_{t-1}+u_t))^2}{\sigma^2}\right\}\right)$
  - Final constraint: $\underset{x,z}{\operatorname{argmin}} \sum \frac{(z_t-(x_t+m_t))^2}{\sigma^2} + \frac{(x_t-(x_{t-1}+u_t))^2}{\sigma^2}$
- Example

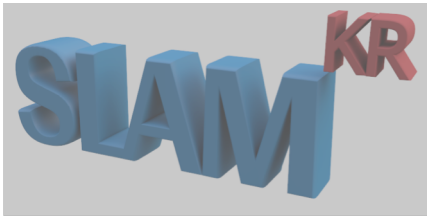


$$J_{GraphSLAM} = \frac{(z_1-7)^2}{\sigma^2} + \frac{(x_1-(x_0+10))^2}{\sigma^2} + \frac{(z_1-(x_1-4))^2}{\sigma^2}$$

- Parameters maximizing the likelihood
  - $z_1 = 6.67, x_1 = 10.33$

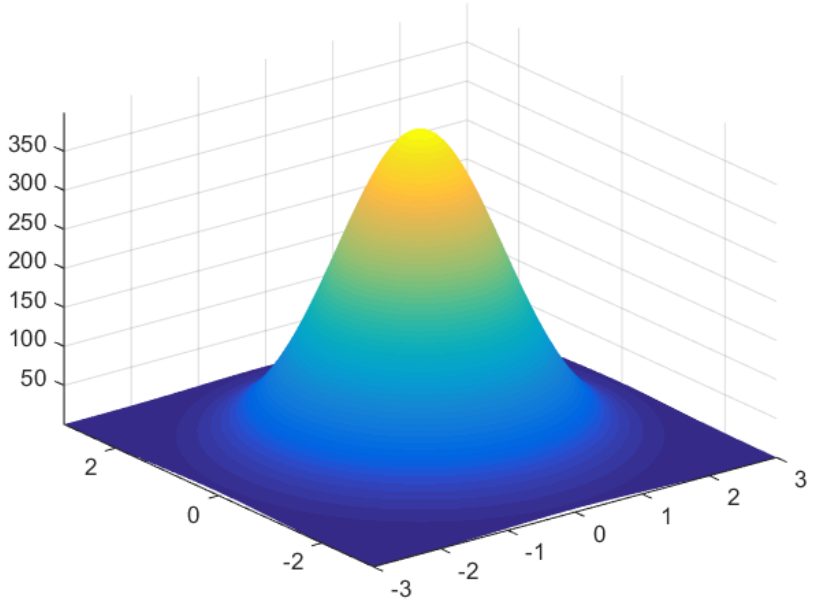
(proof)

n-Dimensional Case



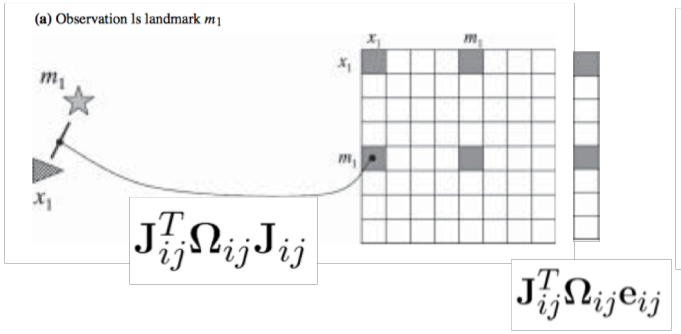
- n-D measurement constraint: $\left(z_t - h(x_t, m_j)\right)^T Q_t^{-1} \left(z_t - h(x_t, m_j)\right)$
- n-D motion constraint: $\left(x_t - g(u_t, x_{t-1})\right)^T R_t^{-1} \left(x_t - g(u_t, x_{t-1})\right)$

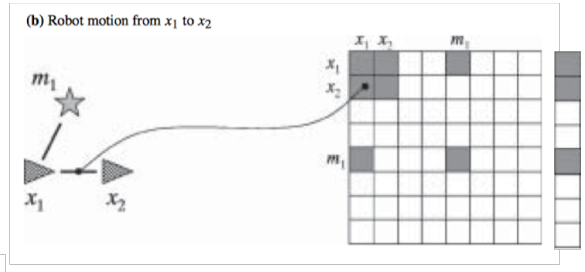
- Multidimensional formula
  - $\arg\min \sum_t \left(x_t - g(u_t, x_{t-1})\right)^T R_t^{-1} \left(x_t - g(u_t, x_{t-1})\right) + \left(z_t - h(x_t, m_j)\right)^T Q_t^{-1} \left(z_t - h(x_t, m_j)\right)$

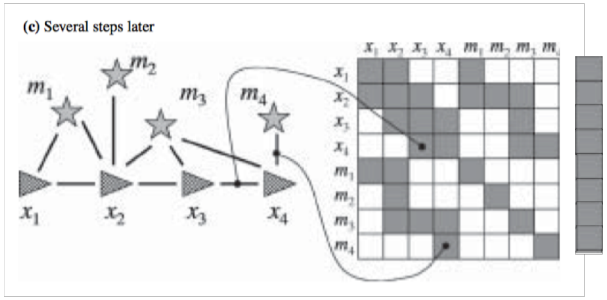


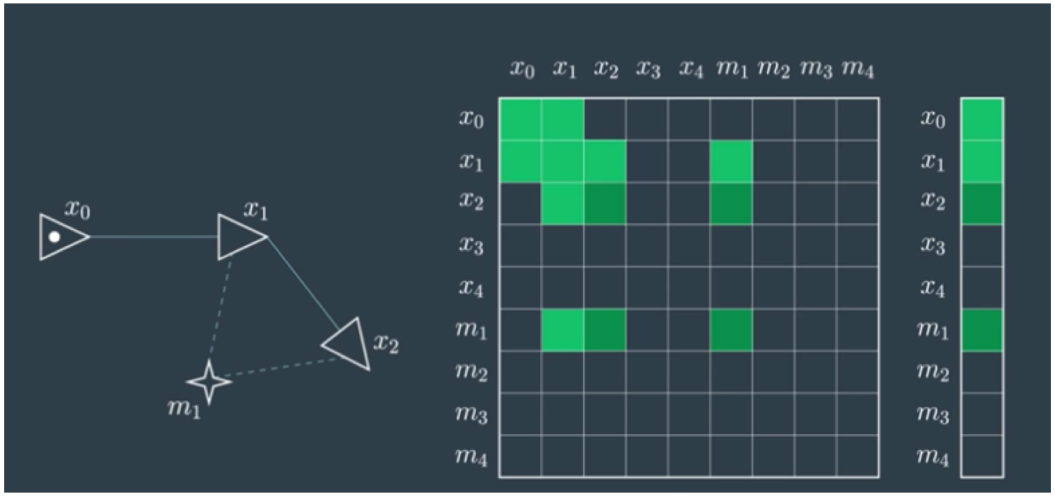
Information Matrix and Vector

- Each measurement and control -> a local update of information matrix and vector









Information matrix H

Information vector b

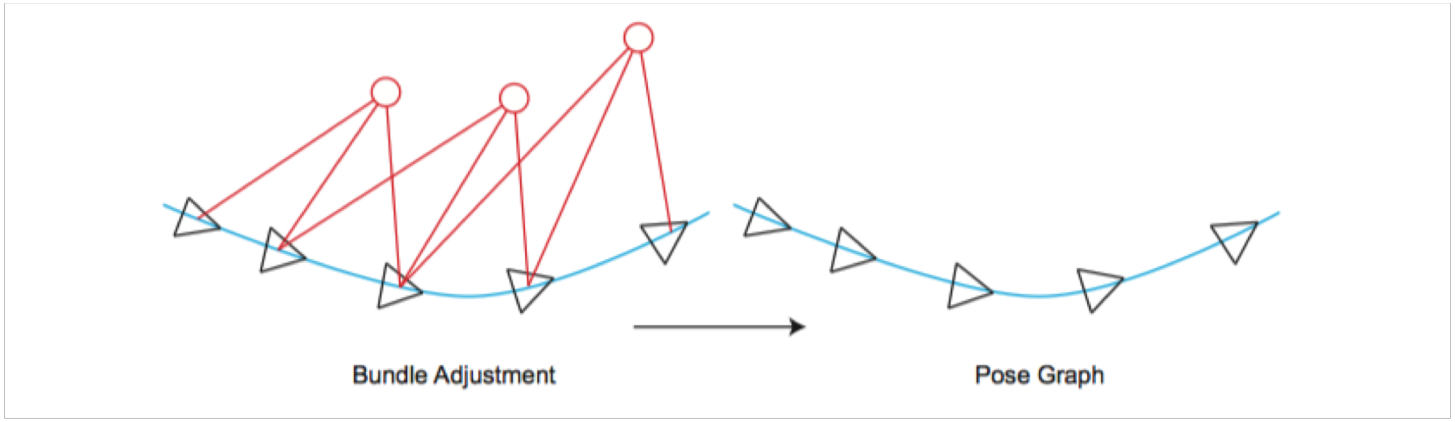
Small increment

$$\Delta \mathbf{x}^* = -\mathbf{H}^{-1}\mathbf{b}$$

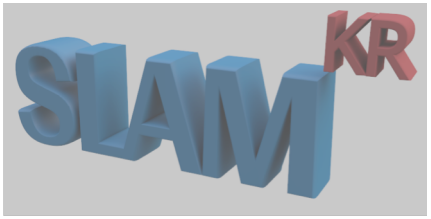
S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," in *International Journal of Robotics Research*, 2006.

Pose Graph Optimizaion (PGO)

- Bundle adjustment, graph slam의 단점
  - 시간이 지남에 따라 로봇의 이동 궤적이 길어지고 지도의 규모가 계속 커짐
  - 지도의 규모가 커짐에 따라 계산해야하는 constraint의 수가 기하급수적으로 늘어남
- 랜드마크를 무시하고 최적화를 수행할수 있을까?
- 로봇의 포즈만 고려하는 그래프 최적화를 구성
- 랜드마크를 이용한 로봇 포즈의 초기추정이 완료되면
  더이상 해당 랜드마크의 위치를 최적화 하지않고 카메라 포즈간의 연결만 최적화
- Constrinats in a pose graph
  - Odometry constraint
  - Loop closure constraint



Least Squares Optimization



- The log likelihood

$$l_{ij} \propto [\underbrace{\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)}_{\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)}]^T \mathbf{\Omega}_{ij} [\mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)].$$

- Least squares optimization

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \underbrace{\mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}}_{\mathbf{F}_{ij}},$$

- Optimize the following equation

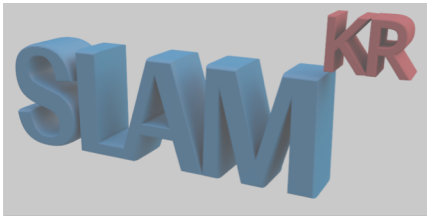
$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}} \mathbf{F}(\mathbf{x}).$$

- First order Taylor expansion

$$\begin{aligned} \mathbf{e}_{ij}(\breve{\mathbf{x}}_i + \mathbf{\Delta x}_i, \breve{\mathbf{x}}_j + \mathbf{\Delta x}_j) &= \mathbf{e}_{ij}(\breve{\mathbf{x}} + \mathbf{\Delta x}) \\ &\simeq \mathbf{e}_{ij} + \mathbf{J}_{ij} \mathbf{\Delta x}. \end{aligned}$$

G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based SLAM,” *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010.

Least Squares Optimization



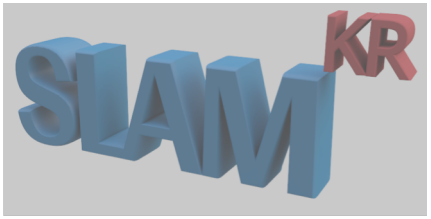
- Substituting the equation

$$
\begin{aligned}
&\mathbf{F}_{ij}(\breve{\mathbf{x}} + \boldsymbol{\Delta}\mathbf{x}) \\
&\quad = \quad \mathbf{e}_{ij}(\breve{\mathbf{x}} + \boldsymbol{\Delta}\mathbf{x})^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}(\breve{\mathbf{x}} + \boldsymbol{\Delta}\mathbf{x}) \\
&\quad \simeq \quad (\mathbf{e}_{ij} + \mathbf{J}_{ij}\boldsymbol{\Delta}\mathbf{x})^T \boldsymbol{\Omega}_{ij} (\mathbf{e}_{ij} + \mathbf{J}_{ij}\boldsymbol{\Delta}\mathbf{x}) \\
&\quad = \quad \underbrace{\mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{e}_{ij}}_{\mathbf{c}_{ij}} + 2 \underbrace{\mathbf{e}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}} \boldsymbol{\Delta}\mathbf{x} + \boldsymbol{\Delta}\mathbf{x}^T \underbrace{\mathbf{J}_{ij}^T \boldsymbol{\Omega}_{ij} \mathbf{J}_{ij}}_{\mathbf{H}_{ij}} \boldsymbol{\Delta}\mathbf{x} \\
&\quad = \quad \mathbf{c}_{ij} + 2\mathbf{b}_{ij}\boldsymbol{\Delta}\mathbf{x} + \boldsymbol{\Delta}\mathbf{x}^T \mathbf{H}_{ij} \boldsymbol{\Delta}\mathbf{x}
\end{aligned}
$$

- Rewrite the function F(x)

$$
\begin{aligned}
\mathbf{F}(\breve{\mathbf{x}} + \boldsymbol{\Delta}\mathbf{x}) \quad &= \quad \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{F}_{ij}(\breve{\mathbf{x}} + \boldsymbol{\Delta}\mathbf{x}) \\
&\simeq \quad \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{c}_{ij} + 2\mathbf{b}_{ij}\boldsymbol{\Delta}\mathbf{x} + \boldsymbol{\Delta}\mathbf{x}^T \mathbf{H}_{ij} \boldsymbol{\Delta}\mathbf{x} \\
&= \quad \mathbf{c} + 2\mathbf{b}^T \boldsymbol{\Delta}\mathbf{x} + \boldsymbol{\Delta}\mathbf{x}^T \mathbf{H} \boldsymbol{\Delta}\mathbf{x}. \quad \text{where } \mathbf{c} = \sum \mathbf{c}_{ij},\ \mathbf{b} = \sum \mathbf{b}_{ij} \text{ and } \mathbf{H} = \sum \mathbf{H}_{ij}.
\end{aligned}
$$

Least Squares Optimization



- Minimizing the quadratic form by,

$$\mathbf{H}\,\Delta\mathbf{x}^{*} \;=\; -\mathbf{b}.$$

- Small increment

$$\Delta\mathbf{x}^{*} = -\mathbf{H}^{-1}\mathbf{b}$$

- The solution is obtained by adding the increments $\Delta x*$ to the initial guess

$$\mathbf{x}^{*} \;=\; \breve{\mathbf{x}} + \Delta\mathbf{x}^{*}.$$

Structure of Linearized System

- Every constraint will contribute to the system with an addend term.
- The structure of this addend depends on the Jacobian of the error function.
- General form of the Jacobian
  - the error function of a constraint depends only on the values of two nodes

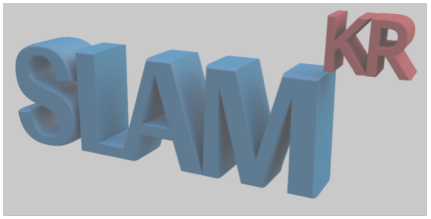
$$\mathbf{J}_{ij} = \begin{pmatrix} \mathbf{0} \cdots \mathbf{0} & \underbrace{\mathbf{A}_{ij}}_{\text{node } i} & \mathbf{0} \cdots \mathbf{0} & \underbrace{\mathbf{B}_{ij}}_{\text{node } j} & \mathbf{0} \cdots \mathbf{0} \end{pmatrix}$$

- The Information matrix H and the information vector b

$$\mathbf{H}_{ij} = \begin{pmatrix} \ddots & & & & \\ & \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij} & \cdots & \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij} & \\ & \vdots & \ddots & \vdots & \\ & \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij} & \cdots & \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij} & \\ & & & & \ddots \end{pmatrix}$$

$$\mathbf{b}_{ij} = \begin{pmatrix} \vdots \\ \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij} \\ \vdots \\ \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij} \\ \vdots \end{pmatrix}$$

Pseudo Code



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 ¬converged **do**

$\mathbf{b} \leftarrow \mathbf{0} \qquad \mathbf{H} \leftarrow \mathbf{0}$

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

$\mathbf{A}_{ij} \leftarrow \frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_i}\Big|_{\mathbf{x}=\breve{\mathbf{x}}} \qquad \mathbf{B}_{ij} \leftarrow \frac{\partial \mathbf{e}_{ij}(\mathbf{x})}{\partial \mathbf{x}_j}\Big|_{\mathbf{x}=\breve{\mathbf{x}}}$

// compute the contribution of this constraint to the linear system

$\mathbf{H}_{[ii]} \mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij} \qquad \mathbf{H}_{[ij]} \mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij}$

$\mathbf{H}_{[ji]} \mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{A}_{ij} \qquad \mathbf{H}_{[jj]} \mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{B}_{ij}$

// compute the coefficient vector

$\mathbf{b}_{[i]} \mathrel{+}= \mathbf{A}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij} \qquad \mathbf{b}_{[j]} \mathrel{+}= \mathbf{B}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}$

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

$\mathbf{x}^* \leftarrow \breve{\mathbf{x}}$

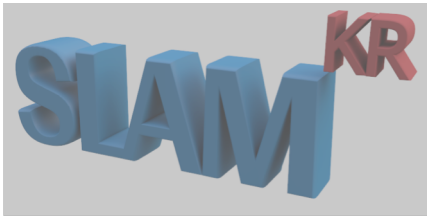
$\mathbf{H}^* \leftarrow \mathbf{H}$

// release the first node

$\mathbf{H}^*_{[11]} \mathrel{-}= \mathbf{I}$

return $\langle \mathbf{x}^*, \mathbf{H}^* \rangle$

PGO on a Manifold



- 전체적인 목적 함수의 형태 (least squares)

$$\min_{\boldsymbol{\xi}} \frac{1}{2} \sum_{i,j \in \mathcal{E}} \boldsymbol{e}_{ij}^{\mathrm{T}} \boldsymbol{\Sigma}_{ij}^{-1} \boldsymbol{e}_{ij}.$$

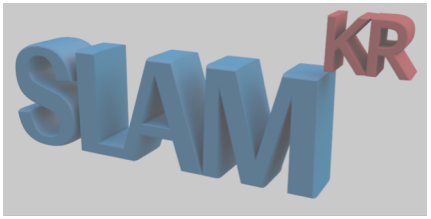
- Residual on a Manifold

$$\begin{aligned} \boldsymbol{e}_{ij} &= \ln \left(\boldsymbol{T}_{ij}^{-1} \boldsymbol{T}_i^{-1} \boldsymbol{T}_j \right)^{\vee} \\ &= \ln \left(\exp((-\boldsymbol{\xi}_{ij})^{\wedge}) \exp((-\boldsymbol{\xi}_i)^{\wedge}) \exp(\boldsymbol{\xi}_j^{\wedge}) \right)^{\vee}. \end{aligned}$$

- 오차값의 로봇 포즈 ξ_i와 ξ_j에 대한 섭동

$$\hat{\boldsymbol{e}}_{ij} = \ln \left(\boldsymbol{T}_{ij}^{-1} \boldsymbol{T}_i^{-1} \exp((-\boldsymbol{\delta\xi}_i)^{\wedge}) \exp(\delta\boldsymbol{\xi}_j^{\wedge}) \boldsymbol{T}_j \right)^{\vee}.$$

PGO on a Manifold



- 테일러 근사와 같은 형태의 식 유도

$$
\begin{aligned}
\hat{\boldsymbol{e}}_{ij} &= \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\exp((-\boldsymbol{\delta\xi}_i)^\wedge)\exp(\boldsymbol{\delta\xi}_j^\wedge)\boldsymbol{T}_j\right)^\vee \\
&= \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\boldsymbol{T}_j\exp\left(\left(-\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_i\right)^\wedge\right)\exp\left(\left(\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_j\right)^\wedge\right)\right)^\vee \\
&\approx \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\boldsymbol{T}_j\left[\boldsymbol{I}-(\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_i)^\wedge+(\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_j)^\wedge\right]\right)^\vee \\
&\approx \boldsymbol{e}_{ij}+\frac{\partial \boldsymbol{e}_{ij}}{\partial\boldsymbol{\delta\xi}_i}\boldsymbol{\delta\xi}_i+\frac{\partial \boldsymbol{e}_{ij}}{\partial\boldsymbol{\delta\xi}_j}\boldsymbol{\delta\xi}_j
\end{aligned}
$$

$\exp(\boldsymbol{\xi}^\wedge)\boldsymbol{T}=\boldsymbol{T}\exp\left(\left(\mathrm{Ad}(\boldsymbol{T}^{-1})\boldsymbol{\xi}\right)^\wedge\right)$. 대치 후,

섭동항을 오른쪽으로 몰아놓음

테일러 근사

????

$$\frac{\partial \boldsymbol{e}_{ij}}{\partial\boldsymbol{\delta\xi}_i}=-\boldsymbol{\mathcal{J}}_r^{-1}(\boldsymbol{e}_{ij})\mathrm{Ad}(\boldsymbol{T}_j^{-1}).$$

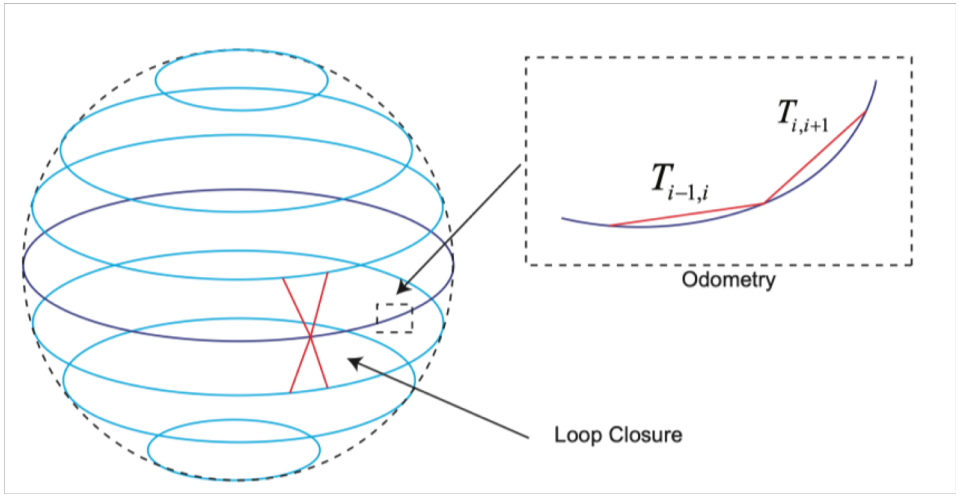
$$\frac{\partial \boldsymbol{e}_{ij}}{\partial\boldsymbol{\delta\xi}_j}=\boldsymbol{\mathcal{J}}_r^{-1}(\boldsymbol{e}_{ij})\mathrm{Ad}(\boldsymbol{T}_j^{-1}).$$

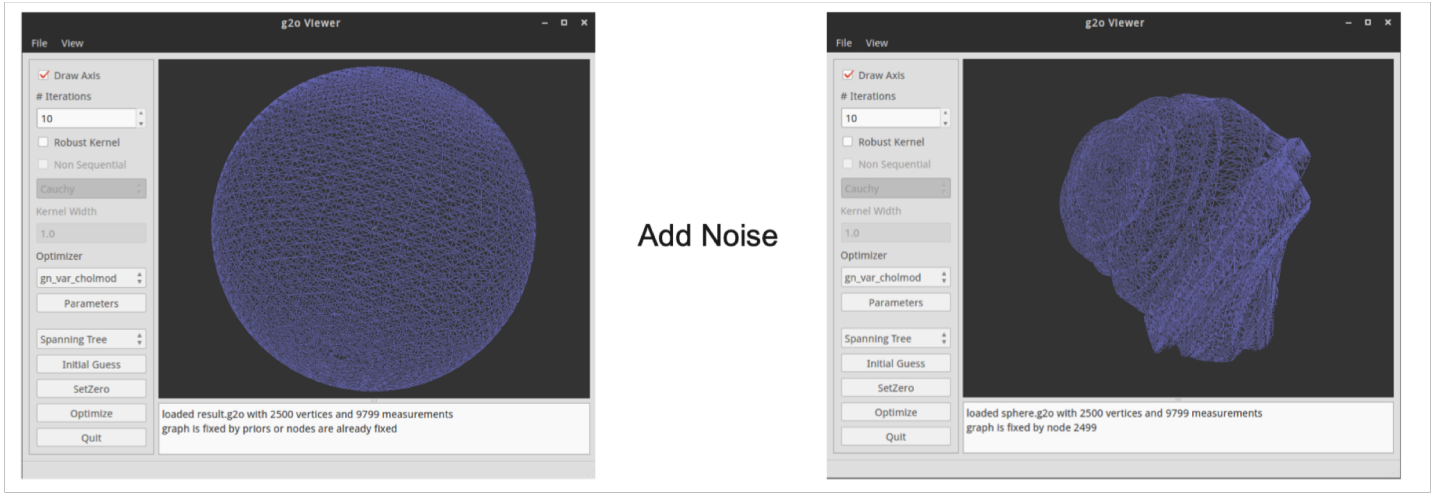
$$\boldsymbol{\mathcal{J}}_r^{-1}(\boldsymbol{e}_{ij})\approx\boldsymbol{I}+\frac{1}{2}\begin{bmatrix}\boldsymbol{\phi}_e^\wedge & \boldsymbol{\rho}_e^\wedge\\ \boldsymbol{0} & \boldsymbol{\phi}_e^\wedge\end{bmatrix}$$

- 파라미터의 변화량에 따른 에러 펑션의 변화량을 알수 있게 된다

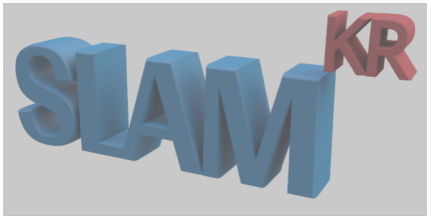
실험

- 구체에 대한 PGO


$T_{i,i+1}$
$T_{i-1,i}$
Odometry
Loop Closure


g2o Viewer
File View
Draw Axis
Iterations
10
Robust Kernel
Non Sequential
Cauchy
Kernel Width
1.0
Optimizer
gn_var_cholmod
Parameters
Spanning Tree
Initial Guess
SetZero
Optimize
Quit
loaded result.g2o with 2500 vertices and 9799 measurements
graph is fixed by priors or nodes are already fixed
Add Noise
g2o Viewer
File View
Draw Axis
Iterations
10
Robust Kernel
Non Sequential
Cauchy
Kernel Width
1.0
Optimizer
gn_var_cholmod
Parameters
Spanning Tree
Initial Guess
SetZero
Optimize
Quit
loaded sphere.g2o with 2500 vertices and 9799 measurements
graph is fixed by node 2499

실험 결과



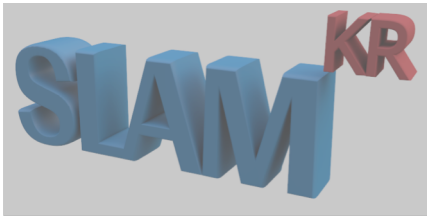
- 최적화 결과 (쿼터니언 사용)

```
$ build/pose_graph_g2o_SE3 sphere.g2o
read total 2500 vertices, 9799 edges.
prepare optimizing ...
calling optimizing ...
iteration= 0  chi2= 1023011093.967638 time= 0.851879 cumTime= 0.851879 edges= 9799 schur= 0 lambda=
     805.622433 levenbergIter= 1
iteration= 1  chi2= 385118688.233188  time= 0.863567 cumTime= 1.71545  edges= 9799 schur= 0 lambda=
     537.081622 levenbergIter= 1
iteration= 2  chi2= 166223726.693659  time= 0.861235 cumTime= 2.57668  edges= 9799 schur= 0 lambda=
     358.054415 levenbergIter= 1
iteration= 3  chi2= 86610874.269316   time= 0.844105 cumTime= 3.42079  edges= 9799 schur= 0 lambda=
     238.702943 levenbergIter= 1
iteration= 4  chi2= 40582782.710190   time= 0.862221 cumTime= 4.28301  edges= 9799 schur= 0 lambda=
     159.135295 levenbergIter= 1
......
iteration= 28 chi2= 45095.174398 time= 0.869451 cumTime= 30.0809 edges= 9799 schur= 0 lambda= 0.003127
      levenbergIter= 1
iteration= 29 chi2= 44811.248504 time= 1.76326  cumTime= 31.8442 edges= 9799 schur= 0 lambda= 0.003785
      levenbergIter= 2
saving optimization results ...
```

- Manifold 상에서의 최적화 결과

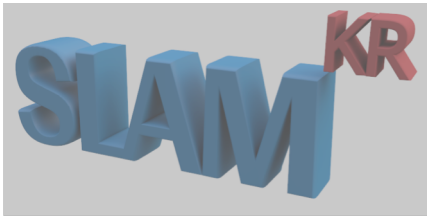
```
loaded result_lie.g2o with 2500 vertices and 9799 measurements
graph is fixed by node 2499
# Using CHOLMOD poseDim -1 landMarkDim -1 blockordering 0
Preparing (no marginalization of Landmarks)
iteration= 0 chi2= 44360.509723 time= 0.567504 cumTime= 0.567504 edges= 9799 schur= 0
iteration= 1 chi2= 44360.471110 time= 0.595993 cumTime= 1.1635   edges= 9799 schur= 0
iteration= 2 chi2= 44360.471110 time= 0.582909 cumTime= 1.74641  edges= 9799 schur= 0
```

Thank you



Appendix

Proof



- $z_1 = 6.67, x_1 = 10.33$

$$
\begin{aligned}
J_{GraphSLAM} &= \frac{(z_1 - (x_0 + 7))^2}{\sigma^2} + \frac{(x_1 - (x_0 + 10))^2}{\sigma^2} + \frac{(z_1 - (x_1 - 4))^2}{\sigma^2} && \sigma^2 \text{ same} \\
&= (z_1 - (x_0 + 7))^2 + (x_1 - (x_0 + 10))^2 + (z_1 - (x_1 - 4))^2 && x_0 = 0 \\
&= (z_1 - 7)^2 + (x_1 - 10)^2 + (z_1 - (x_1 - 4))^2 \\
&= (z_1^2 - 14z_1 + 49) + (x_1^2 - 20x_1 + 100) + (x^2 + z_1^2 + 16 - 8x_1 + 8z_1 - 2x_1z_1) \\
&= 2z_1^2 + 2x_1^2 - 28x_1 - 6z_1 - 2x_1z_1 + 165
\end{aligned}
$$

$J_{GraphSLAM} = 2z_1^2 + 2x_1^2 - 28x_1 - 6z_1 - 2x_1z_1 + 165$

Derivative with respect to z_1 :

$$4z_1 - 6 - 2x_1 = 0$$

Derivative with respect to x_1 :

$$4x_1 - 28 - 2z_1 = 0$$

Solve the system of equations:

$$
\begin{aligned}
4z_1 - 2x_1 &= 6 \\
+ \quad -4z_1 + 8x_1 &= 56 \\
\hline
6x_1 &= 62 \\
x_1 &= 62/6 \\
&= 10.\bar{3}
\end{aligned}
$$

Substitute the value for x_1
into any equation to calculate the value for z_1 :

$$
\begin{aligned}
4z_1 - 2x_1 &= 6 \\
4z_1 - 2(62/6) &= 6 \\
4z_1 &= 80/3 \\
z_1 &= 20/3 \\
&= 6.\bar{6}
\end{aligned}
$$