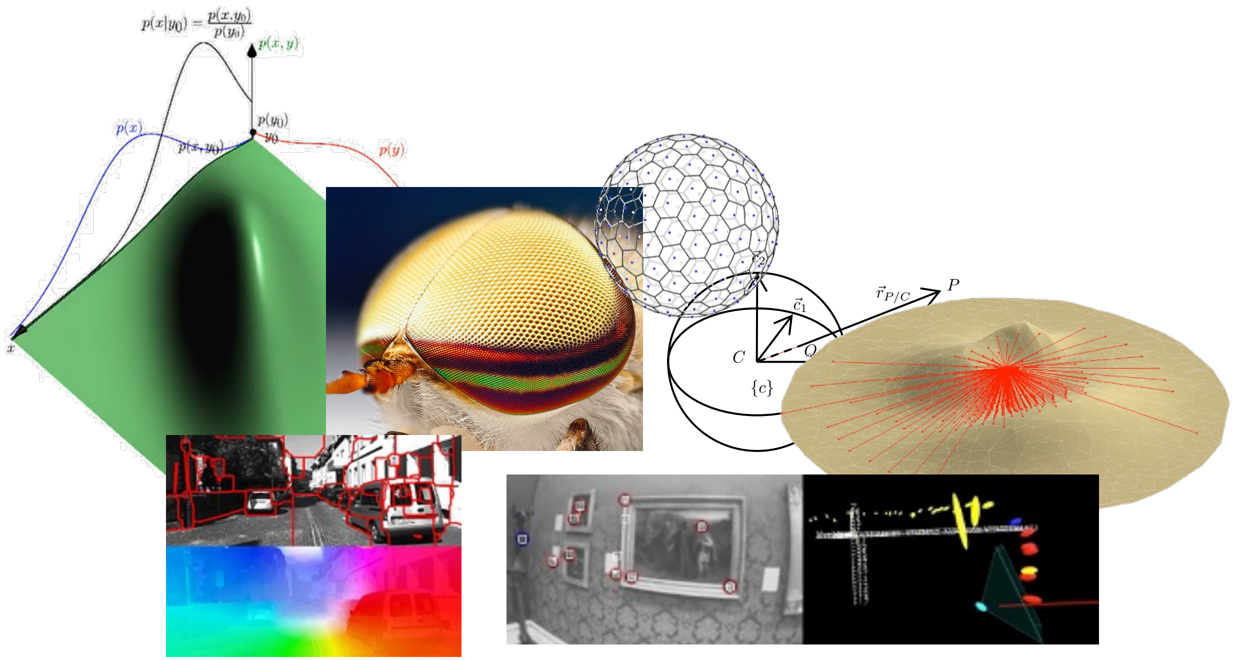
MCHA4400

Vision-based navigation



Dr Chris Renton

School of Engineering

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

FACULTY OF ENGINEERING AND BUILT ENVIRONMENT

www.newcastle.edu.au

Features vs. landmarks

A **feature** is a point or region of interest in an image, e.g.,

- Corners
- Regions
- AR tags

A **landmark** is a point or object of interest in the world, e.g.,

- Points
- Extended objects
- Rigid bodies

A feature is a potential measurement of a landmark.

There may be features in an image that do not correspond to any landmark.

There may be landmarks that do not have a corresponding feature in the image.

State vector

To perform simultaneous localisation and mapping (SLAM) using landmarks, we augment the vehicle state (pose and pose velocity) with landmark states,

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\eta} \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}$$

$$\boldsymbol{\nu} = \begin{bmatrix} \mathbf{v}^b_{B/N} \\ \boldsymbol{\omega}^b_{\mathcal{B}/\mathcal{N}} \end{bmatrix}$$

$$\mathbf{T}^n_b = \begin{bmatrix} \mathbf{R}^n_b & \mathbf{r}^n_{B/N} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \in \mathsf{SE}(3) \qquad \mathbf{R}^n_b = \mathbf{R}(\boldsymbol{\Theta}^n_b)$$

$$\boldsymbol{\eta} = \begin{bmatrix} \mathbf{r}^n_{B/N} \\ \boldsymbol{\Theta}^n_b \end{bmatrix}$$

where $\mathbf{m}_j$ are the states associated with the j^{th} landmark.

State distribution

The state distribution is the joint Gaussian distribution of the velocity, pose and map

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\eta} \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{\nu}} \\ \boldsymbol{\mu}_{\boldsymbol{\eta}} \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \vdots \\ \boldsymbol{\mu}_n \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{\boldsymbol{\nu\nu}} & \mathbf{P}_{\boldsymbol{\nu\eta}} & \mathbf{P}_{\boldsymbol{\nu}1} & \mathbf{P}_{\boldsymbol{\nu}2} & \cdots & \mathbf{P}_{\boldsymbol{\nu}n} \\ \mathbf{P}_{\boldsymbol{\eta\nu}} & \mathbf{P}_{\boldsymbol{\eta\eta}} & \mathbf{P}_{\boldsymbol{\eta}1} & \mathbf{P}_{\boldsymbol{\eta}2} & \cdots & \mathbf{P}_{\boldsymbol{\eta}n} \\ \mathbf{P}_{1\boldsymbol{\nu}} & \mathbf{P}_{1\boldsymbol{\eta}} & \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1n} \\ \mathbf{P}_{2\boldsymbol{\nu}} & \mathbf{P}_{2\boldsymbol{\eta}} & \mathbf{P}_{21} & \mathbf{P}_{22} & \cdots & \mathbf{P}_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{n\boldsymbol{\nu}} & \mathbf{P}_{n\boldsymbol{\eta}} & \mathbf{P}_{n1} & \mathbf{P}_{n2} & \cdots & \mathbf{P}_{nn} \end{bmatrix}\right)$$

Process model

For the vehicle states, we may assume body-fixed acceleration driven by a zero-mean Gaussian process.

We assume the landmark states are constant.

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{\nu}}(t) \\ \dot{\boldsymbol{\eta}}(t) \\ \dot{\mathbf{m}}_1(t) \\ \dot{\mathbf{m}}_2(t) \\ \vdots \\ \dot{\mathbf{m}}_n(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{J}_K(\boldsymbol{\eta}(t))\,\boldsymbol{\nu}(t) \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\mathbf{f}(\mathbf{x}(t))} + \underbrace{\begin{bmatrix} \dot{\mathbf{w}}_{\boldsymbol{\nu}}(t) \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\dot{\mathbf{w}}(t)}$$

$$\dot{\mathbf{w}}_{\boldsymbol{\nu}}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_{\boldsymbol{\nu}}\,\delta(t - t'))$$

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Point landmarks

A *point landmark* is represented by the position of a point w.r.t. a world reference point expressed in world coordinates, i.e.,

$$\mathbf{m}_j = \mathbf{r}^n_{j/N} \in \mathbb{R}^3$$

for each landmark index j.

Let $\mathbf{y}_i \in \mathbb{R}^2$ be the pixel coordinates of the measured point feature that corresponds to the j^{th} point landmark.

Then we may propose the following Gaussian measurement likelihood:

$$p(\mathbf{y}_i|\mathbf{x}) = \mathcal{N}(\mathbf{y}_i; \mathtt{w2p}(\mathbf{m}_j; \mathbf{T}^n_b, \boldsymbol{\theta}), \mathbf{R})$$

where `w2p` is the `worldToPixel` function.

Pose landmarks

A *pose landmark* is represented by the parameters describing the position and orientation of a rigid body with respect to the world reference point and coordinate system, i.e.,

$$\mathbf{T}_j^n = \begin{bmatrix} \mathbf{R}_j^n & \mathbf{r}_{j/N}^n \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \in \mathsf{SE}(3) \qquad \mathbf{R}_j^n = \mathbf{R}(\boldsymbol{\Theta}_j^n)$$

$$\mathbf{m}_j = \begin{bmatrix} \mathbf{r}_{j/N}^n \\ \boldsymbol{\Theta}_j^n \end{bmatrix}$$

where the j^{th} landmark has a reference point j and a local coordinate system $\{j\}$.

Pose landmarks can be measured with three or more non-collinear point features that are associated with that landmark.

Landmark measurements

Assuming independence, the joint measurement likelihood for all potentially visible landmarks in the current frame can be written as the product of conditionally independent potentially visible landmark measurement likelihoods,

$$p(\mathbf{y}|\mathbf{x}) = \prod_{(i,j)\in\mathcal{A}} p_{\mathcal{A}}(\mathbf{y}_i|\boldsymbol{\eta}, \mathbf{m}_j) \prod_{j\in\mathcal{U}} p_{\mathcal{U}}(\mathbf{y}_\emptyset|\boldsymbol{\eta}, \mathbf{m}_j)$$

where $\mathcal{A}$ is the set of associated feature/landmark pairs, $\mathcal{U}$ is the set of potentially visible landmarks not associated with any feature, $p(\mathbf{y}_i|\boldsymbol{\eta}, \mathbf{m}_j)$ is the likelihood for landmark j, which is known to be associated with feature i, and

$$p_{\mathcal{U}}(\mathbf{y}_\emptyset|\boldsymbol{\eta}, \mathbf{m}_j) = \mathcal{U}(\mathbf{y}; \mathcal{Y}) = \begin{cases} \frac{1}{|\mathcal{Y}|} & \text{if } \mathbf{y} \in \mathcal{Y} \\ 0 & \text{if } \mathbf{y} \notin \mathcal{Y} \end{cases}$$

is the likelihood for landmark j, which is not associated with any feature (uniform distribution over the entire image).

"potentially visible" = within camera FOV

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Landmark measurements

The log-likelihood is given by

$$
\begin{aligned}
\log p(\mathbf{y}|\mathbf{x}) &= \log \prod_{(i,j)\in\mathcal{A}} p_{\mathcal{A}}(\mathbf{y}_i|\boldsymbol{\eta},\mathbf{m}_j) \prod_{j\in\mathcal{U}} p_{\mathcal{U}}(\mathbf{y}_{\emptyset}|\boldsymbol{\eta},\mathbf{m}_j) \\
&= \sum_{(i,j)\in\mathcal{A}} \log p_{\mathcal{A}}(\mathbf{y}_i|\boldsymbol{\eta},\mathbf{m}_j) + \sum_{j\in\mathcal{U}} \log p_{\mathcal{U}}(\mathbf{y}_{\emptyset}|\boldsymbol{\eta},\mathbf{m}_j) \\
&= \sum_{(i,j)\in\mathcal{A}} \log p_{\mathcal{A}}(\mathbf{y}_i|\boldsymbol{\eta},\mathbf{m}_j) + \sum_{j\in\mathcal{U}} \log \frac{1}{|\mathcal{Y}|} \\
&= \sum_{(i,j)\in\mathcal{A}} \log p_{\mathcal{A}}(\mathbf{y}_i|\boldsymbol{\eta},\mathbf{m}_j) - |\mathcal{U}| \log |\mathcal{Y}|
\end{aligned}
$$

where $|\mathcal{U}|$ is the number of potentially visible landmarks not associated with any feature and $|\mathcal{Y}|$ is the image area in pixel units.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Landmark measurements

Within the Laplace information filter, the measurement update can be performed as follows:

$$
\begin{aligned}
\mathcal{V}(\mathbf{x}) &= -\sum_{(i,j)\in\mathcal{A}} \log p_{\mathcal{A}}(\mathbf{y}_i|\boldsymbol{\eta}, \mathbf{m}_j) + |\mathcal{U}| \log|\mathcal{Y}| - \log \mathcal{N}^{-\frac{1}{2}}(\mathbf{x}; \boldsymbol{\nu}_{\mathbf{x}}, \boldsymbol{\Xi}_{\mathbf{x}}) \\
\boldsymbol{\mu}_{\mathbf{x}} &= \boldsymbol{\Xi}_{\mathbf{x}}^{-1}\boldsymbol{\nu}_{\mathbf{x}} \\
(\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}, \boldsymbol{\Xi}_{\mathbf{x}|\mathbf{y}}) &= \textsc{fmin}\text{BFGS}\textsc{TrustSqrt}(\mathcal{V}(\cdot), \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Xi}_{\mathbf{x}}) \\
\boldsymbol{\nu}_{\mathbf{x}|\mathbf{y}} &= \boldsymbol{\Xi}_{\mathbf{x}|\mathbf{y}}\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}
\end{aligned}
$$

where $\boldsymbol{\Xi}_{\mathbf{x}|\mathbf{y}}$ is upper triangular.

Note: The association set $\mathcal{A}$ and number of unassociated landmarks, $|\mathcal{U}|$, depend on the state, $\mathbf{x}$.

Map management

Landmark initialisation

Consider a state with two landmarks

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\eta} \\ \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_\nu \\ \boldsymbol{\mu}_\eta \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{\nu\nu} & \mathbf{P}_{\nu\eta} & \mathbf{P}_{\nu 1} & \mathbf{P}_{\nu 2} \\ \mathbf{P}_{\eta\nu} & \mathbf{P}_{\eta\eta} & \mathbf{P}_{\eta 1} & \mathbf{P}_{\eta 2} \\ \mathbf{P}_{1\nu} & \mathbf{P}_{1\eta} & \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{2\nu} & \mathbf{P}_{2\eta} & \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}\right)$$

When we add a third landmark, i.e.,

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\eta} \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_\nu \\ \boldsymbol{\mu}_\eta \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_3 \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{\nu\nu} & \mathbf{P}_{\nu\eta} & \mathbf{P}_{\nu 1} & \mathbf{P}_{\nu 2} & \mathbf{P}_{\nu 3} \\ \mathbf{P}_{\eta\nu} & \mathbf{P}_{\eta\eta} & \mathbf{P}_{\eta 1} & \mathbf{P}_{\eta 2} & \mathbf{P}_{\eta 3} \\ \mathbf{P}_{1\nu} & \mathbf{P}_{1\eta} & \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} \\ \mathbf{P}_{2\nu} & \mathbf{P}_{2\eta} & \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} \\ \mathbf{P}_{3\nu} & \mathbf{P}_{3\eta} & \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} \end{bmatrix}\right)$$

how do we populate the new mean and covariance entries?

Landmark initialisation

Since we initialise a new landmark from its first measurement, the new joint mean and covariance should be consistent with that measurement.

Therefore, let us consider the resulting state distribution following a measurement update.

Since all of the information about the new landmark should come from the measurement, we can consider the following prior state distribution:

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\eta} \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix} ; \begin{bmatrix} \boldsymbol{\mu}_\nu \\ \boldsymbol{\mu}_\eta \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_3 \end{bmatrix} , \begin{bmatrix} \mathbf{P}_{\nu\nu} & \mathbf{P}_{\nu\eta} & \mathbf{P}_{\nu 1} & \mathbf{P}_{\nu 2} & \mathbf{0} \\ \mathbf{P}_{\eta\nu} & \mathbf{P}_{\eta\eta} & \mathbf{P}_{\eta 1} & \mathbf{P}_{\eta 2} & \mathbf{0} \\ \mathbf{P}_{1\nu} & \mathbf{P}_{1\eta} & \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{0} \\ \mathbf{P}_{2\nu} & \mathbf{P}_{2\eta} & \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \infty\mathbf{I} \end{bmatrix} \right)$$

Since the uncertainty in $\mathbf{m}_3$ is infinite, the value assigned to $\boldsymbol{\mu}_3$ shouldn't matter. This prior represents our belief on the new landmark position before it is observed.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Landmark initialisation

Let the joint camera velocity, camera pose and landmark distribution be given by a Gaussian distribution in square-root information form as follows:

$$p(\mathbf{x}) = \mathcal{N}^{-\frac{1}{2}}(\mathbf{x}; \boldsymbol{\nu}, \boldsymbol{\Xi})$$

To insert a new landmark $\mathbf{m}_+$ at the end of the state, we assume that the prior on $\mathbf{m}_+$ is independent and uninformative and construct the following distribution

$$\begin{aligned} p(\mathbf{x}, \mathbf{m}_+) &= p(\mathbf{x})\, p(\mathbf{m}_+) \\ &= \mathcal{N}^{-\frac{1}{2}}(\mathbf{x}; \boldsymbol{\nu}, \boldsymbol{\Xi})\, \mathcal{N}^{-\frac{1}{2}}(\mathbf{m}_+; \mathbf{0}, \mathbf{0}) \\ &= \mathcal{N}^{-\frac{1}{2}}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{m}_+ \end{bmatrix}; \begin{bmatrix} \boldsymbol{\nu} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Xi} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\right) \end{aligned}$$

Then, we perform the usual measurement update, which completes the initialisation.

See also `GaussianInfo::operator*=`.

Resizing

Beware resizing vectors and matrices if you care about their contents.

$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$$

`resize(4, 4)`

$$\begin{bmatrix}1&5&9&\cdot\\4&8&\cdot&\cdot\\7&3&\cdot&\cdot\\2&6&\cdot&\cdot\end{bmatrix} \text{ or } \begin{bmatrix}\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\\\cdot&\cdot&\cdot&\cdot\end{bmatrix}$$

`conservativeResize(4, 4)`

$$\begin{bmatrix}1&2&3&\cdot\\4&5&6&\cdot\\7&8&9&\cdot\\\cdot&\cdot&\cdot&\cdot\end{bmatrix}$$

```
conservativeResizeLike(
    Eigen::MatrixXd::Zero(4, 4))
```

$$\begin{bmatrix}1&2&3&0\\4&5&6&0\\7&8&9&0\\0&0&0&0\end{bmatrix}$$

Elements denoted with “·” are uninitialised.

Landmark initialisation

Recall the cost function used in the LIF measurement update,

$$\mathcal{V}(\mathbf{x}) = \frac{n}{2}\log 2\pi - \sum_{i=1}^{n} \log |\boldsymbol{\Xi}|_{ii} + \frac{1}{2}\left\|\boldsymbol{\Xi}\mathbf{x} - \boldsymbol{\nu}\right\|^2 - \sum_{(i,j)\in\mathcal{A}} \log p_{\mathcal{A}}(\mathbf{y}_i|\mathbf{x}) + |\mathcal{U}| \log |\mathcal{Y}|$$

When we augment the prior square-root information vector and matrix with the new landmark prior, the cost becomes

$$\begin{aligned}\mathcal{V}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{m}_+\end{bmatrix}\right) &= \frac{n+n_+}{2}\log 2\pi - \sum_{i=1}^{n} \log |\boldsymbol{\Xi}|_{ii} + \frac{1}{2}\left\|\begin{bmatrix}\boldsymbol{\Xi} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\nu}\\ \mathbf{0}\end{bmatrix}\right\|^2 - \sum_{(i,j)\in\mathcal{A}} \log p_{\mathcal{A}}(\mathbf{y}_i|\mathbf{x},\mathbf{m}_+) + |\mathcal{U}| \log |\mathcal{Y}| \\ &= \frac{n+n_+}{2}\log 2\pi - \sum_{i=1}^{n} \log |\boldsymbol{\Xi}|_{ii} + \frac{1}{2}\left\|\boldsymbol{\Xi}\mathbf{x} - \boldsymbol{\nu}\right\|^2 - \sum_{(i,j)\in\mathcal{A}} \log p_{\mathcal{A}}(\mathbf{y}_i|\mathbf{x},\mathbf{m}_+) + |\mathcal{U}| \log |\mathcal{Y}|\end{aligned}$$

Since we can safely evaluate the negative log prior before augmentation, we can do the augmentation within the measurement update itself.

We can also force the new landmark, $\mathbf{m}_+$, to be associated with the feature used to initialise it, $\mathbf{y}_+$.

Landmark initialisation

If the measurement of a new landmark does not provide enough information to initialise its depth from the camera (e.g., point landmarks), the optimisation may fail due to a singular Hessian matrix.

A simple workaround is to initialise new landmarks using

$$\begin{aligned} p(\mathbf{x}, \mathbf{m}_+) &= p(\mathbf{x})\, p(\mathbf{m}_+) \\ &= \mathcal{N}^{-\frac{1}{2}}(\mathbf{x}; \boldsymbol{\nu}, \boldsymbol{\Xi})\, \mathcal{N}^{-\frac{1}{2}}(\mathbf{m}_+; \boldsymbol{\nu}_+, \boldsymbol{\Xi}_+) \\ &= \mathcal{N}^{-\frac{1}{2}}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{m}_+ \end{bmatrix}; \begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\nu}_+ \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Xi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Xi}_+ \end{bmatrix}\right) \end{aligned}$$

$$\begin{aligned} \boldsymbol{\Xi}_+ &= \epsilon \mathbf{I} \\ \boldsymbol{\nu}_+ &= \boldsymbol{\Xi}_+ \boldsymbol{\mu}_+ \end{aligned}$$

where the prior mean of the new landmark $\boldsymbol{\mu}_+$ is set somewhere within the camera field of view at an arbitrary depth* and ϵ is a small value such that $1/\epsilon$ supports the range of expected depth. The subsequent measurement update will then collapse the prior spherical confidence region into a narrow ellipsoid.

*This choice effectively sets the map scale in monocular SLAM with point landmarks.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Landmark initialisation

For example, to set the position of a new point landmark at depth *d* along the camera optical axis, let

$$\vec{r}_{+/N} = \vec{r}_{C/N} + d\,\vec{c}_3$$
$$\mathbf{r}^n_{+/N} = \mathbf{r}^n_{C/N} + d\,\mathbf{c}^n_3$$

$$\mathbf{R}^n_c = \begin{bmatrix} \mathbf{c}^n_1 & \mathbf{c}^n_2 & \mathbf{c}^n_3 \end{bmatrix}$$

or to set the position of a new point landmark at depth *d* along the measured ray, let

$$\vec{r}_{+/N} = \vec{r}_{C/N} + d\,\vec{u}_{P/C}$$
$$\mathbf{r}^n_{+/N} = \mathbf{r}^n_{C/N} + d\,\mathbf{R}^n_c \mathbf{u}^c_{P/C}$$
$$\mathbf{r}^n_{+/N} = \mathbf{r}^n_{C/N} + d\,\mathbf{R}^n_c\,\mathrm{pixelToVector}(\mathbf{y}_+)$$

$$\vec{u}_{P/C} = \frac{\vec{r}_{P/C}}{\|\vec{r}_{P/C}\|}$$

Then an affine transform can be used to initialise the new point landmark as follows:

$$\boldsymbol{\Omega}\colon \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} \mapsto \mathbf{r}^n_{+/N} + \mathbf{e} \qquad \mathbf{e} \sim \mathcal{N}^{-\frac{1}{2}}(\mathbf{x}; \mathbf{0}, \epsilon\mathbf{I})$$

$$\mathcal{T}^{-\frac{1}{2}}\{\boldsymbol{\Omega}\}\colon \left(\begin{bmatrix} \boldsymbol{\nu} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Xi} & \mathbf{0} \\ \mathbf{0} & \epsilon\mathbf{I} \end{bmatrix} \right) \mapsto (\boldsymbol{\nu}_+, \boldsymbol{\Xi}_+)$$

Landmark initialisation with quasi-Newton optimisation

The iterated measurement update uses the Laplace approximation, which sets the posterior information matrix to the Hessian. If the Hessian is being approximated using a quasi-Newton method, there may be situations where optimisation successfully converges to the correct solution, but the Hessian is not accurate.

When the number of iterations taken to find the optimum solution is smaller than the dimension of the measurement, not enough Hessian updates are performed to obtain a good approximation. This can occur immediately following landmark initialisation in SLAM if the landmark state mean is initialised very close to the posterior solution. In this case, the optimisation may terminate in 1 or 2 iterations, without correctly determining the landmark state uncertainty.

Workarounds:

- Don’t initialise landmarks exactly at known solution.
- Post-calculate the Hessian whenever initialising new landmarks.
- Use Newton optimisation whenever initialising new landmarks.

Landmark deletion

Since we have finite computing resources, there is an upper limit to the number of landmarks, N_{max}, that can be stored in the map and still attain real-time operation.

Some landmarks are better than others—they can be consistently re-identified from different perspectives. These are the type of landmarks that we want to keep in the map.

A landmark in the map may not even correspond to a real position in space. This can occur if it has been initialised from a corner feature that is the intersection of edges at different depths. Such landmarks will fail to be re-identified and should be removed from the map.

A useful heuristic is that after 10 consecutive failures to match the predicted feature location with the measured feature location, the landmark should be deleted.

Landmark deletion

Consider the following state distribution

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\eta} \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_\nu \\ \boldsymbol{\mu}_\eta \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_3 \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{\nu\nu} & \mathbf{P}_{\nu\eta} & \mathbf{P}_{\nu 1} & \mathbf{P}_{\nu 2} & \mathbf{P}_{\nu 3} \\ \mathbf{P}_{\eta\nu} & \mathbf{P}_{\eta\eta} & \mathbf{P}_{\eta 1} & \mathbf{P}_{\eta 2} & \mathbf{P}_{\eta 3} \\ \mathbf{P}_{1\nu} & \mathbf{P}_{1\eta} & \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} \\ \mathbf{P}_{2\nu} & \mathbf{P}_{2\eta} & \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} \\ \mathbf{P}_{3\nu} & \mathbf{P}_{3\eta} & \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} \end{bmatrix}\right)$$

To remove landmark 2, simply remove its corresponding element in the mean and row and column in the covariance,

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\nu} \\ \boldsymbol{\eta} \\ \mathbf{m}_1 \\ \mathbf{m}_3 \end{bmatrix}; \begin{bmatrix} \boldsymbol{\mu}_\nu \\ \boldsymbol{\mu}_\eta \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_3 \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{\nu\nu} & \mathbf{P}_{\nu\eta} & \mathbf{P}_{\nu 1} & \mathbf{P}_{\nu 3} \\ \mathbf{P}_{\eta\nu} & \mathbf{P}_{\eta\eta} & \mathbf{P}_{\eta 1} & \mathbf{P}_{\eta 3} \\ \mathbf{P}_{1\nu} & \mathbf{P}_{1\eta} & \mathbf{P}_{11} & \mathbf{P}_{13} \\ \mathbf{P}_{3\nu} & \mathbf{P}_{3\eta} & \mathbf{P}_{31} & \mathbf{P}_{33} \end{bmatrix}\right)$$

Landmark deletion is equivalent to marginalisation.

See also `GaussianInfo::marginal`.

Landmark SLAM overview

Start with initial state with no landmarks.

For each camera frame,

1. Perform time update to current frame time
2. Identify landmarks with matching features
3. Remove failed landmarks from map (consecutive failures to match)
4. Identify surplus features that do not correspond to landmarks in the map
5. Initialise up to $N_{\text{max}} - N$ new landmarks from best surplus features
6. Perform measurement update

Post-processing

Marginal distributions

Given the joint distribution,

$$p(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\nu}\\ \boldsymbol{\eta}\\ \mathbf{m}_1\\ \mathbf{m}_2\\ \mathbf{m}_3\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_{\boldsymbol{\nu}}\\ \boldsymbol{\mu}_{\boldsymbol{\eta}}\\ \boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\\ \boldsymbol{\mu}_3\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{\boldsymbol{\nu}\boldsymbol{\nu}} & \mathbf{P}_{\boldsymbol{\nu}\boldsymbol{\eta}} & \mathbf{P}_{\boldsymbol{\nu}1} & \mathbf{P}_{\boldsymbol{\nu}2} & \mathbf{P}_{\boldsymbol{\nu}3}\\ \mathbf{P}_{\boldsymbol{\eta}\boldsymbol{\nu}} & \mathbf{P}_{\boldsymbol{\eta}\boldsymbol{\eta}} & \mathbf{P}_{\boldsymbol{\eta}1} & \mathbf{P}_{\boldsymbol{\eta}2} & \mathbf{P}_{\boldsymbol{\eta}3}\\ \mathbf{P}_{1\boldsymbol{\nu}} & \mathbf{P}_{1\boldsymbol{\eta}} & \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13}\\ \mathbf{P}_{2\boldsymbol{\nu}} & \mathbf{P}_{2\boldsymbol{\eta}} & \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23}\\ \mathbf{P}_{3\boldsymbol{\nu}} & \mathbf{P}_{3\boldsymbol{\eta}} & \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33}\end{bmatrix}\right)$$

then, the marginal distributions of the camera velocity, camera pose and landmarks are typically of interest,

$$\begin{aligned}
p(\boldsymbol{\nu}) &= \mathcal{N}(\boldsymbol{\nu}; \boldsymbol{\mu}_{\boldsymbol{\nu}}, \mathbf{P}_{\boldsymbol{\nu}\boldsymbol{\nu}})\\
p(\boldsymbol{\eta}) &= \mathcal{N}(\boldsymbol{\eta}; \boldsymbol{\mu}_{\boldsymbol{\eta}}, \mathbf{P}_{\boldsymbol{\eta}\boldsymbol{\eta}})\\
p(\mathbf{m}_1) &= \mathcal{N}(\mathbf{m}_1; \boldsymbol{\mu}_1, \mathbf{P}_{11})\\
p(\mathbf{m}_2) &= \mathcal{N}(\mathbf{m}_2; \boldsymbol{\mu}_2, \mathbf{P}_{22})\\
p(\mathbf{m}_3) &= \mathcal{N}(\mathbf{m}_3; \boldsymbol{\mu}_3, \mathbf{P}_{33})
\end{aligned}$$

See also `GaussianInfo::marginal`.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Marginal confidence regions

Once we have the marginal distributions of the camera pose and landmark positions, we can compute and plot their confidence regions.

Recall the confidence region that encloses probability mass c is given by

$$\mathcal{R} = \{\mathbf{x} \in \mathcal{X} \mid p(\mathbf{x}) \geq t\}$$

where the density threshold, t, is the unique solution to

$$\int_{p(\mathbf{x}) \geq t} p(\mathbf{x}) \, \mathrm{d}\mathbf{x} = c$$

Gaussian confidence region

The confidence region of an n-dimensional Gaussian pdf

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P}) = \mathcal{N}^{-\frac{1}{2}}(\mathbf{x}; \boldsymbol{\nu}, \boldsymbol{\Xi})$$

that encloses a given probability mass $0 < c < 1$ is the n-dimensional hyperellipsoid defined by points **x** that satisfy the following inequality:

$$(\mathbf{x} - \boldsymbol{\mu})^{\mathsf{T}} \mathbf{P}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq \chi_n^2(c)$$

$$(\boldsymbol{\Xi}\mathbf{x} - \boldsymbol{\nu})^{\mathsf{T}} (\boldsymbol{\Xi}\mathbf{x} - \boldsymbol{\nu}) \leq \chi_n^2(c)$$

where the term on the right is the inverse cumulative distribution function for probability c of the chi-squared distribution with n degrees of freedom.

In MATLAB, this can be computed as follows:

$$\chi_n^2(c) = \texttt{chi2inv}(c, n)$$

Confidence ellipsoids in 3 dimensions

In three dimensions, the boundary of the ellipsoid is given by

$$(\boldsymbol{\Xi}\mathbf{x} - \boldsymbol{\nu})^{\mathsf{T}}(\boldsymbol{\Xi}\mathbf{x} - \boldsymbol{\nu}) = \chi_3^2(c)$$

which can be compactly written in homogeneous coordinates

$$\mathbf{p}^{\mathsf{T}}\mathbf{Q}\mathbf{p} = 0$$

where

$$\mathbf{p} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} \boldsymbol{\Xi}^{\mathsf{T}}\boldsymbol{\Xi} & -\boldsymbol{\Xi}^{\mathsf{T}}\boldsymbol{\nu} \\ -\boldsymbol{\nu}^{\mathsf{T}}\boldsymbol{\Xi} & \boldsymbol{\nu}^{\mathsf{T}}\boldsymbol{\nu} - \chi_3^2(c) \end{bmatrix}$$

The matrix $\mathbf{Q}$ contains the *quadric surface coefficients*, which can be efficiently rendered in GLU, VTK, etc.

Confidence ellipses in the image

To compute the confidence region of predicted feature locations in the image, we can propagate the marginal distribution of the landmark position through the measurement model with an affine transform.

$$\boldsymbol{\Psi}\colon \begin{bmatrix}\mathbf{x}\\ \mathbf{v}\end{bmatrix} \mapsto \mathbf{y} = \mathbf{h}_j(\mathbf{x}) + \mathbf{v}$$

$$\mathcal{T}^{-\frac{1}{2}}\{\boldsymbol{\Psi}\}\colon \left(\begin{bmatrix}\boldsymbol{\nu}_\mathbf{x}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Xi}_\mathbf{xx} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Xi}_\mathbf{R}\end{bmatrix}\right) \mapsto (\boldsymbol{\nu}_\mathbf{y}, \boldsymbol{\Xi}_\mathbf{yy})$$

$$\mathbf{h}_j(\mathbf{x}) = \texttt{w2p}(\mathbf{m}_j; \mathbf{T}^n_b, \boldsymbol{\theta})$$
$$\mathbf{v} \sim \mathcal{N}^{-\frac{1}{2}}(\mathbf{0}, \boldsymbol{\Xi}_\mathbf{R})$$

We can then plot the confidence *ellipse* in the image,

$$(\boldsymbol{\Xi}_\mathbf{yy}\mathbf{y} - \boldsymbol{\nu}_\mathbf{y})^\mathsf{T}(\boldsymbol{\Xi}_\mathbf{yy}\mathbf{y} - \boldsymbol{\nu}_\mathbf{y}) = \mathbf{w}^\mathsf{T}\mathbf{w} = \chi^2_2(c)$$

where $\mathbf{w} = \boldsymbol{\Xi}_\mathbf{yy}\mathbf{y} - \boldsymbol{\nu}_\mathbf{y}$

In $\mathbf{w}$-coordinates, the boundary of the confidence region is a circle with radius $\sqrt{\chi^2_2(c)}$

Generate points on a circle in $\mathbf{w}$-coordinates then solve the triangular system of equations

$$\boldsymbol{\Xi}_\mathbf{yy}\mathbf{y} = \mathbf{w} + \boldsymbol{\nu}_\mathbf{y}$$

to obtain points on the boundary of the confidence ellipse.

Standard deviations to probability mass

The probability mass, c, contained within n_σ standard deviations of the mean of a Gaussian pdf is given by

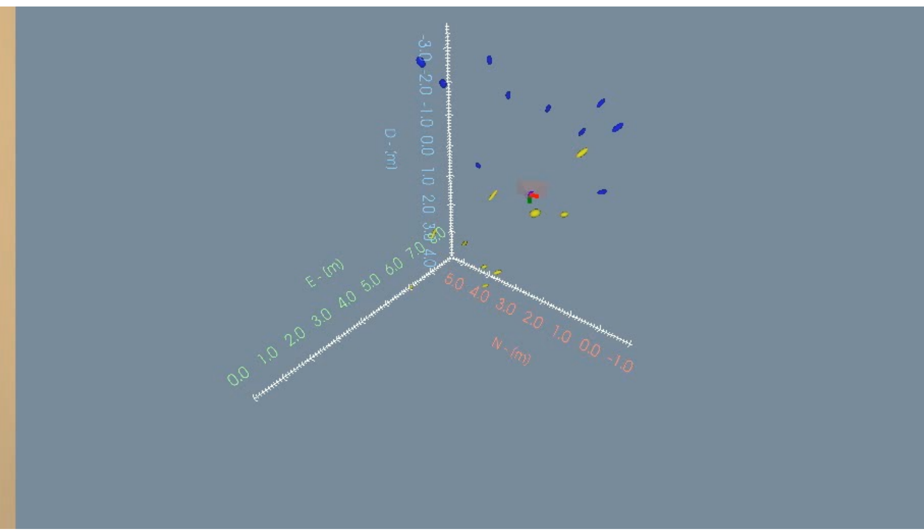
$$c = \int_{\mu - n_\sigma \sigma}^{\mu + n_\sigma \sigma} \mathcal{N}(x; \mu, \sigma^2)\,\mathrm{d}x = \int_{-n_\sigma}^{n_\sigma} \mathcal{N}(x; 0, 1)\,\mathrm{d}x$$

This can be integrated numerically or computed from the normal cumulative distribution function.

For example, in MATLAB, you can use `normcdf`, which computes

$$\texttt{normcdf}(x; \mu, \sigma) = \int_{-\infty}^{x} \mathcal{N}(z; \mu, \sigma^2)\,\mathrm{d}z$$

Example — Unique tag SLAM and loop closure



References

Early EKF SLAM:

- Davison, A. J., Murray, D. W., 2002. Simultaneous localization and map-building using active vision. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (7), 865–880.

Undelayed landmark initialisation:

- Montiel, J.M., Civera, J. and Davison, A.J., 2006, August. Unified inverse depth parametrization for monocular SLAM. Robotics: Science and Systems.
- Civera, J., Davison, A. J., Montiel, J., 2008. Inverse depth parametrization for monocular SLAM. Robotics, IEEE Transactions on 24 (5), 932–945.

Summary and review papers:

- Marzorati, D., Matteucci, M., Migliore, D., Sorrenti, D. G., 2009. On the use of inverse scaling in monocular SLAM. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, pp. 2030–2036.
- Sola, J., Vidal-Calleja, T., Civera, J., Montiel, J. M. M., 2012. Impact of landmark parametrization on monocular EKF-SLAM with points and lines. International journal of computer vision 97 (3), 339–368.