Intrinsic parameter estimation on Lie groups
Applications in mapping and localization from a wearable camera

Guillaume Bourmaud, PhD Candidate
University of Bordeaux, IMS Laboratory, Signal & Image Group

27/07/15
• PhD funded by dem@care integrated European project (2012-2015)
Context

- PhD funded by dem@care integrated European project (2012-2015)

- Objective ⇒ Indoor localization of the patient from a wearable camera
Illustration of the problem

Video frame examples

Part I
How to automatically build an annotated database?

Part II
Assuming the DB is given, how to localize a new video?
Illustration of the problem

Video frame examples

Annotated Database (DB)
Illustration of the problem

Video frame examples

Annotated Database (DB)

Assuming the DB is given, how to localize a new video? ⇒ Part I
Illustration of the problem

Video frame examples

Annotated Database (DB)

Assuming the DB is given, how to localize a new video? ⇒ Part I

How to automatically build an annotated database? ⇒ Part II
Part 1

Visual Indoor Localization
Visual Indoor Localization Problem

Difficult problem: illumination changes, moving objects, blur, environment changes between video to localize and DB...
Content Based Image Retrieval (CBIR) [1]

Good retrieval

Wrong retrieval

Example of estimated trajectory for a video sequence

Ground truth

CBIR (NN)
Example of estimated trajectory for a video sequence

**Ground truth**

**CBIR Post-processing**

**CBIR (NN)**
Example of estimated trajectory for a video sequence

Ground truth

CBIR Post-processing

trajectory smoothness

= Smoother
Example of estimated trajectory for a video sequence

Ground truth

CBIR (NN)

CBIR Post-processing

- trajectory smoothness
  + outlier matches

= Particle Smoother
Example of estimated trajectory for a video sequence

Ground truth

CBIR (NN)

---

**CBIR Post-processing**

- trajectory smoothness
- outlier matches
- efficiency

= Marginalized Particle Smoother
Example of estimated trajectory for a video sequence

Ground truth

CBIR (NN)

CBIR Post-processing

- trajectory smoothness
- outlier matches
- efficiency

= Marginalized Particle Smoother

This work
Motivation

Marginalized particle smoother (a.k.a Rao-Blackwellized particle smoother)

- Sample discrete variable ⇒ inlier/outlier classification
- For each classification, estimate a trajectory ⇒ Iterated Extended Kalman Filter (IEKF) + Rauch-Tung-Striebel smoother (RTS)
- Take the centroid of the trajectories as estimate
Motivation

Marginalized particle smoother (a.k.a Rao-Blackwellized particle smoother)

- Sample discrete variable $\Rightarrow$ inlier/outlier classification
- For each classification, estimate a trajectory $\Rightarrow$ Iterated Extended Kalman Filter (IEKF) + Rauch-Tung-Striebel smoother (RTS)
- Take the centroid of the trajectories as estimate

Problem $\Rightarrow$ camera pose evolves on $SE(3)$

Not a Euclidean space!
Motivation

Marginalized particle smoother (a.k.a Rao-Blackwellized particle smoother)

- Sample discrete variable $\Rightarrow$ inlier/outlier classification
- For each classification, estimate a trajectory $\Rightarrow$ Iterated Extended Kalman Filter (IEKF) + Rauch-Tung-Striebel smoother (RTS)
- Take the centroid of the trajectories as estimate

**Problem** $\Rightarrow$ camera pose evolves on $SE(3)$

Not a Euclidean space!

How to take *intrinsically* into account the constraints of $SE(3)$?
Motivation

Marginalized particle smoother (a.k.a Rao-Blackwellized particle smoother)

- Sample discrete variable $\Rightarrow$ inlier/outlier classification
- For each classification, estimate a trajectory $\Rightarrow$ Iterated Extended Kalman Filter (IEKF) + Rauch-Tung-Striebel smoother (RTS)
- Take the centroid of the trajectories as estimate

Problem $\Rightarrow$ camera pose evolves on $SE(3)$

Not a Euclidean space!

How to take *intrinsically* into account the constraints of $SE(3)$?

$\Rightarrow$ Generalization of IEKF and RTS to Lie groups
One way to generalize the IEKF/RTS to Lie groups

IEKF/RTS estimation $\Rightarrow$ Bayesian framework $\Rightarrow$ Posterior distribution estim.
One way to generalize the IEKF/RTS to Lie groups

IEKF/RTS estimation ⟹ Bayesian framework ⟹ Posterior distribution estim.

<table>
<thead>
<tr>
<th>Posterior distribution assumption</th>
<th>Fitting method</th>
<th>Mode finding method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean space</td>
<td>Multivariate normal</td>
<td>Gauss-Laplace Approx.</td>
</tr>
<tr>
<td>Lie group</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Part 1: Outline

1. Introduction
2. Preliminaries on matrix Lie groups
3. Estimation tools (probability distribution, optimization,...)
4. Iterated Extended Kalman Filter on Lie Groups
5. Application to localization
Part 1: Outline

1. Introduction

2. Preliminaries on matrix Lie groups

3. Estimation tools (probability distribution, optimization,...)

4. Iterated Extended Kalman Filter on Lie Groups

5. Application to localization
Matrix Lie group $G$

$G \subset \mathbb{R}^{n \times n}$ = smooth manifold + matrix group

Solution: $X(t) = \exp_{G}(a t)$

where $\exp_{G}: T_{\text{Id}}G \rightarrow G$
Introduction to Lie groups

Matrix Lie group $G$

Lie group $G \subset \mathbb{R}^{n \times n} = \text{smooth manifold} + \text{matrix group}$

One parameter subgroup

$$\dot{X}(t) = X(t) \; a \; X(0) = \text{Id}$$

where $X(t) \in G$, $a \in T_{\text{Id}G}$, $\dot{X}(t) \in T_{X(t)}G$

Solution:

$$X(t) = \exp^G(a \; t)$$

where $\exp^G: T_{\text{Id}G} \rightarrow G$

Intrinsic parameter estimation on Lie groups: Applications in mapping and localization from a wearable camera

27/07/15

8 / 36
Introduction to Lie groups

Matrix Lie group $G$

$G \subset \mathbb{R}^{n \times n} =$ smooth manifold + matrix group

Solution:

\[
X(t) = \exp_{G}(a t)
\]

where $\exp_{G}: T_{Id}G \rightarrow G$
Introduction to Lie groups

Matrix Lie group $G$

Lie group $G \subset \mathbb{R}^{n \times n} = \text{smooth manifold} + \text{matrix group}$

One parameter subgroup

\[
\dot{X}(t) = X(t) a \\
X(0) = \text{Id}
\]

where $X(t) \in G, a \in T_{\text{Id}} G, \dot{X}(t) \in T_{X(t)} G$

Solution:

$X(t) = \exp_{G}(a t)$

where $\exp_{G} : T_{\text{Id}} G \rightarrow G$

Intrinsic parameter estimation on Lie groups: Applications in mapping and localization from a wearable camera 27/07/15
**Matrix Lie group $G$**

Lie group $G \subset \mathbb{R}^{n \times n} = $ smooth manifold + matrix group

**One parameter subgroup**

\[
\begin{align*}
\dot{X}(t) &= X(t) a \\
X(0) &= I_d
\end{align*}
\]

where $X(t) \in G$, $a \in T_{Id}G$, $\dot{X}(t) \in T_{X(t)}G$
Introduction to Lie groups

Matrix Lie group $G$

$G \subset \mathbb{R}^{n \times n} = \text{smooth manifold } + \text{matrix group}$

One parameter subgroup

\[
\begin{align*}
\dot{X}(t) &= X(t)a \\
X(0) &= \text{Id}
\end{align*}
\]

where $X(t) \in G$, $a \in T_{\text{Id}}G$,

$\dot{X}(t) \in T_{X(t)}G$

Solution: $X(t) = \exp_G(at)$ where

$\exp_G : T_{\text{Id}}G \to G$

Lie Group Manifold $G \subset \mathbb{R}^{n \times n}$
Introduction to Lie groups

Lie algebra $\mathfrak{g}$

tangent space at the identity $T_{Id}G = \text{Lie algebra } \mathfrak{g} \subset \mathbb{R}^{n \times n}$
Introduction to Lie groups

Lie algebra $\mathfrak{g}$

**tangent space at the identity** $T_{Id}G = \text{Lie algebra } \mathfrak{g} \subset \mathbb{R}^{n \times n}$

$\text{Lie Algebra } \mathfrak{g} = T_{Id}G$

$\text{Lie Group Manifold } G \subset \mathbb{R}^{n \times n}$
Introduction to Lie groups

Lie algebra \( g \)

Tangent space at the identity \( T_{Id}G = \text{Lie algebra } g \subset \mathbb{R}^{n \times n} \)

\( \exp_G(\cdot) \) and \( \log_G(\cdot) \) are bijective in the neighborhood of \( Id \) only!
Introduction to Lie groups

Lie algebra $\mathfrak{g}$

The tangent space at the identity $T_{Id}G = \text{Lie algebra } \mathfrak{g} \subset \mathbb{R}^{n \times n}$

$\exp_G(\cdot)$ and $\log_G(\cdot)$ are bijective in the neighborhood of $Id$ only!
Introduction to Lie groups

Lie algebra \( \mathfrak{g} \)

- tangent space at the identity \( T_{Id} G = \text{Lie algebra } \mathfrak{g} \subset \mathbb{R}^{n \times n} \)

\[
\exp_G(\cdot) = \exp_G([a]_G^\wedge) \\
\log_G^\vee(X) = [\log_G(X)]_G^\vee
\]

\( \exp_G(\cdot) \) and \( \log_G(\cdot) \) are bijective in the neighborhood of \( Id \) only!
Part 1: Outline

1. Introduction

2. Preliminaries on matrix Lie groups

3. Estimation tools (probability distribution, optimization,...)
   - Concentrated Gaussian Distribution on Lie groups
   - Intrinsic Gauss-Laplace Approximation on Lie groups
   - Intrinsic Gauss-Newton Algorithm on Lie groups

4. Iterated Extended Kalman Filter on Lie Groups

5. Application to localization
Part 1: Outline

3 Estimation tools (probability distribution, optimization,...)
   Concentrated Gaussian Distribution on Lie groups
   Intrinsic Gauss-Laplace Approximation on Lie groups
   Intrinsic Gauss-Newton Algorithm on Lie groups
Definition of the Concentrated Gaussian Distribution on Lie groups (CGD)

\[ X \sim \mathcal{N}_G^L (X; \mu, P) \quad \text{if} \quad X = \mu \exp_G^\wedge (\epsilon) \]

\[ X \sim \mathcal{N}_G^R (X; \mu, P) \quad \text{if} \quad X = \exp_G^\wedge (\epsilon) \mu \]

\[ \epsilon \sim \mathcal{N}_{\mathbb{R}^p} (\epsilon; 0_{p \times 1}, P), \ P \subset \mathbb{R}^{p \times p} \text{ is a symmetric positive-definite matrix.} \]
Definition of the Concentrated Gaussian Distribution on Lie groups (CGD)

\[ X \sim \mathcal{N}_G^L (X; \mu, P) \quad \text{if} \quad X = \mu \exp_G^\wedge (\epsilon) \]

\[ X \sim \mathcal{N}_G^R (X; \mu, P) \quad \text{if} \quad X = \exp_G^\wedge (\epsilon) \mu \]

\[ \epsilon \sim \mathcal{N}_{\mathbb{R}^p} (\epsilon; 0_{p \times 1}, P), \quad P \subset \mathbb{R}^{p \times p} \text{ is a symmetric positive-definite matrix.} \]
Definition of the Concentrated Gaussian Distribution on Lie groups (CGD)

$$X \sim \mathcal{N}_G^L (X; \mu, P) \quad \text{if} \quad X = \mu \exp^G (\epsilon)$$

$$X \sim \mathcal{N}_G^R (X; \mu, P) \quad \text{if} \quad X = \exp^G (\epsilon) \mu$$

$$\epsilon \sim \mathcal{N}_{\mathbb{R}^p} (\epsilon; 0_{p \times 1}, P), \ P \subset \mathbb{R}^{p \times p} \text{ is a symmetric positive-definite matrix.}$$
**Definition of the Concentrated Gaussian Distribution on Lie groups (CGD)**

\[ X \sim \mathcal{N}_G^L (X; \mu, P) \quad \text{if} \quad X = \mu \exp^G (\epsilon) \]

\[ X \sim \mathcal{N}_G^R (X; \mu, P) \quad \text{if} \quad X = \exp^G (\epsilon) \mu \]

\[ \epsilon \sim \mathcal{N}_{\mathbb{R}^p} (\epsilon; 0_{p \times 1}, P), \quad P \subset \mathbb{R}^{p \times p} \text{ is a symmetric positive-definite matrix.} \]
Definition of the Concentrated Gaussian Distribution on Lie groups (CGD)

\[ X \sim \mathcal{N}_G^L (X; \mu, P) \quad \text{if} \quad X = \mu \exp_G^\wedge (\epsilon) \]

\[ X \sim \mathcal{N}_G^R (X; \mu, P) \quad \text{if} \quad X = \exp_G^\wedge (\epsilon) \mu \]

\[ \epsilon \sim \mathcal{N}_{\mathbb{R}^p} (\epsilon; 0_{p \times 1}, P), \quad P \subset \mathbb{R}^{p \times p} \text{ is a symmetric positive-definite matrix.} \]
Covariance reparametrization

\[ X = \exp_G^\wedge (\epsilon_b) \mu_b \]
Covariance reparametrization

\[ X = \exp_G^\wedge (\epsilon_b) \mu_b = \exp_G^\wedge (\epsilon_b) \mu_b \mu_a^{-1} \mu_a \]
Covariance reparametrization

\[ X = \exp^G_\varsigma(\epsilon_b) \mu_b = \exp^G_\varsigma(\epsilon_b) \mu_b \mu_a^{-1} \mu_a \]

\[ \approx \exp^G_\varsigma(\log^\nu (\mu_b \mu_a^{-1}) + \varphi_G (\log^\nu (\mu_b \mu_a^{-1})) \epsilon_b) \mu_a \]

where \( \Phi_G \mapsto \varphi_G \mapsto -1 \) is the Left Jacobian of \( G \).
Covariance reparametrization

\[ X = \exp_G^\wedge (\epsilon_b) \mu_b = \exp_G^\wedge (\epsilon_b) \mu_b \mu_a^{-1} \mu_a \]
\[ \approx \exp_G^\wedge \left( \log_G^\vee \left( \mu_b \mu_a^{-1} \right) + \varphi_G \left( \log_G^\vee \left( \mu_b \mu_a^{-1} \right) \right) \epsilon_b \right) \mu_a \]

where \( \Phi_G (\cdot) = \varphi_G (\cdot)^{-1} \) is the Left Jacobian of \( G \).
Expression of the probability density of $X$

\[
\int_{\mathbb{R}^p} p(\epsilon) \, d\epsilon = \int_{\mathbb{R}^p} \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\epsilon\|^2_P} \, d\epsilon
\]
Expression of the probability density of $X$

$$\int \mathbb{R}^p p(\epsilon) d_L \epsilon = \int \mathbb{R}^p \frac{1}{\sqrt{(2\pi)^p \det (P)}} e^{-\frac{1}{2} \|\epsilon\|_P^2} d_L \epsilon$$

Change of variables: $\epsilon = \log^V G (\mu^{-1} X)$ then $d_H X = \det (\Phi_G (-\epsilon)) d_L \epsilon$. 
Expression of the probability density of $X$

$$
\int_{\mathbb{R}^p} p(\epsilon) d_L \epsilon = \int_{\mathbb{R}^p} \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\epsilon\|^2_P} d_L \epsilon
$$

Change of variables: $\epsilon = \log^\vee (\mu^{-1}X)$ then $d_H X = \det (\Phi_G (-\epsilon)) d_L \epsilon$.

Consequently, if $\max \left( \operatorname{eig}(P) \right)$ is sufficiently small, we have:

$$
\int_{\mathbb{R}^p} p(\epsilon) d_L \epsilon \approx \int_G \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\log_G^\vee (\mu^{-1}X)\|^2_P} \frac{1}{\det (\Phi_G (-\log^\vee_G (\mu^{-1}X)))} d_H X
$$

$$
= \int_G p(X) d_H X
$$

where

$$
p(X) = \alpha(X) e^{-\frac{1}{2} \|\log_G^\vee (\mu^{-1}X)\|^2_P}
$$
Expression of the probability density of $X$

\[
\int_{\mathbb{R}^p} p(\epsilon) \, d_L \epsilon = \int_{\mathbb{R}^p} \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\epsilon\|^2} \, d_L \epsilon
\]

Change of variables: $\epsilon = \log_G^\vee (\mu^{-1}X)$ then $d_H X = \det(\Phi_G(-\epsilon)) \, d_L \epsilon$.

Consequently, if $\max(\text{eig}(P))$ is sufficiently small, we have:

\[
\int_{\mathbb{R}^p} p(\epsilon) \, d_L \epsilon \approx \int_G \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\log_G^\vee (\mu^{-1}X)\|^2} \frac{1}{\det(\Phi_G(-\log_G^\vee (\mu^{-1}X)))} \, d_H X
\]

\[
= \int_G p(X) \, d_H X
\]

where

\[
p(X) = \alpha(X) \, e^{-\frac{1}{2} \|\log_G^\vee (\mu^{-1}X)\|^2}
\]

and

\[
\alpha(X) = \frac{1}{\sqrt{(2\pi)^p \det \left( \Phi_G(-\log_G^\vee (\mu^{-1}X)) \, P \Phi_G(-\log_G^\vee (\mu^{-1}X))^T \right)}} \approx \frac{1}{\sqrt{(2\pi)^p \det(P)}}
\]
Expression of the probability density of $X$

\[
\int_{\mathbb{R}^p} p(\epsilon) dL\epsilon = \int_{\mathbb{R}^p} \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\epsilon\|^2_P} dL\epsilon
\]

Change of variables: $\epsilon = \log^\vee (\mu^{-1}X)$ then $dHX = \det(\Phi_G(-\epsilon)) dL\epsilon$.

Consequently, if $\max(\text{eig}(P))$ is sufficiently small, we have:

\[
\int_{\mathbb{R}^p} p(\epsilon) dL\epsilon \approx \int_{G} \frac{1}{\sqrt{(2\pi)^p \det(P)}} e^{-\frac{1}{2} \|\log^\vee (\mu^{-1}X)\|^2_P} \frac{1}{\det(\Phi_G(-\log^\vee (\mu^{-1}X)))} dHX
\]

\[
= \int_{G} p(X) dHX
\]

where

\[
p(X) = \alpha(X) e^{-\frac{1}{2} \|\log^\vee (\mu^{-1}X)\|^2_P}
\]

and

\[
\alpha(X) = \frac{1}{\sqrt{(2\pi)^p \det \left( \Phi_G(-\log^\vee (\mu^{-1}X)) P \Phi_G(-\log^\vee (\mu^{-1}X))^T \right)}} \approx \frac{1}{\sqrt{(2\pi)^p \det(P)}}
\]

$\Rightarrow$ Same distribution as Wang et al., “Error Propagation on the Euclidean Group With Applications to Manipulators Kinematics”, IEEE TRO 2006
Example: 2D Special Euclidean Group $SE(2)$

- represents the position and the attitude of an object in a 2D plane
- $SE(2) = \left\{ C = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \mid R \in SO(2), \ t \in \mathbb{R}^2 \right\}$ and
  $se(2) = \left\{ \begin{bmatrix} \theta \\ u \end{bmatrix} \in \mathbb{R}^3 \mid \begin{bmatrix} \theta \\ u \end{bmatrix} \right\}^{\wedge} = \begin{bmatrix} 0 & -\theta & u_x \\ \theta & 0 & u_y \\ 0 & 0 & 0 \end{bmatrix}$
Example: 2D Special Euclidean Group $SE(2)$

- represents the position and the attitude of an object in a 2D plane
- $SE(2) = \left\{ C = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \right| R \in SO(2), t \in \mathbb{R}^2 \right\}$ and

$$se(2) = \left\{ \begin{bmatrix} \theta \\ u \end{bmatrix} \in \mathbb{R}^3 \left| \begin{bmatrix} \theta \\ u \end{bmatrix} \right\rangle_{SE(2)} = \begin{bmatrix} 0 & -\theta & u_x \\ \theta & 0 & u_y \\ 0 & 0 & 0 \end{bmatrix} \right\}$$

- $C = \exp_{SE(2)}(\epsilon) \mu$ where $\epsilon \sim \mathcal{N}_{\mathbb{R}^3}(\epsilon; 0_3 \times 1, P)$,

$$P = \begin{bmatrix} 0.1 & 0.5 & -0.2 \\ 0.5 & 4 & -1 \\ -0.2 & -1 & 1 \end{bmatrix}, \mu = \exp_{SE(2)}\left( \begin{bmatrix} \pi \\ 30 \\ 20 \end{bmatrix} \right) = \begin{bmatrix} -1 & 0 & -12.73 \\ 0 & -1 & 19.09 \\ 0 & 0 & 1 \end{bmatrix}$$
3 Estimation tools (probability distribution, optimization,...)
Concentrated Gaussian Distribution on Lie groups
Intrinsic Gauss-Laplace Approximation on Lie groups
Intrinsic Gauss-Newton Algorithm on Lie groups
Let us consider the following probability density of $X \in G$ where

$\phi : G \rightarrow \mathbb{R}^m$

$$p(X) = \alpha e^{-\|\phi(X)\|_\Sigma^2}$$
Let us consider the following probability density of $X \in G$ where $\phi : G \to \mathbb{R}^m$

$$p(X) = \alpha e^{-\|\phi(X)\|^2_{\Sigma}}$$

One way to fit a CGD $\Rightarrow$ **Intrinsic Gauss-Laplace Approx**
Let us consider the following probability density of $X \in G$ where
\[ \phi : G \rightarrow \mathbb{R}^m \]

\[ p(X) = \alpha e^{-\|\phi(X)\|^2_{\Sigma}} \]

One way to fit a CGD $\Rightarrow$ **Intrinsic Gauss-Laplace Approx**

\[ \hat{X} = \arg\min_{X \in G} \|\phi(X)\|^2_{\Sigma} \]
• Let us consider the following probability density of $X \in G$ where $\phi : G \rightarrow \mathbb{R}^m$

$$p(X) = \alpha e^{-\|\phi(X)\|^2_{\Sigma}}$$

• One way to fit a CGD $\Rightarrow$ **Intrinsic Gauss-Laplace Approx**

1. $
\hat{X} = \arg\min_{X \in G} \|\phi(X)\|^2_{\Sigma}
$

2. $\phi(X) = \phi(\exp_G^\wedge(\delta) \hat{X}) \approx \phi(\hat{X}) + J\delta$

where $J$ is the Lie group Jacobian of $\phi$

$$J = \left. \frac{d\phi(\exp_G^\wedge(s) \hat{X})}{ds} \right|_{s=0}$$
Let us consider the following probability density of $X \in G$ where $\phi : G \to \mathbb{R}^m$

$$p(X) = \alpha e^{-\|\phi(X)\|^2_\Sigma}$$

One way to fit a CGD $\Rightarrow$ **Intrinsic Gauss-Laplace Approx**

1. $$\hat{X} = \arg\min_{X \in G} \|\phi(X)\|^2_\Sigma$$

2. $$\phi(X) = \phi(\exp^G(\delta) \hat{X}) \approx \phi(\hat{X}) + J\delta$$

   where $J$ is the Lie group Jacobian of $\phi$

   $$J = \left. \frac{d\phi(\exp^G(s) \hat{X})}{ds} \right|_{s=0}$$

3. $$q(X) = \mathcal{N}_G^R(X; \hat{X}, P = (J^T \Sigma^{-1} J)^{-1})$$
Estimation tools (probability distribution, optimization,...)

Concentrated Gaussian Distribution on Lie groups
Intrinsic Gauss-Laplace Approximation on Lie groups
Intrinsic Gauss-Newton Algorithm on Lie groups
Gauss-Newton Algorithm on Matrix Lie groups

- Let us consider the following problem:

\[ \hat{X} = \arg \min_{X \in G} \| \phi (X) \|^2_\Sigma \]

where \( \phi (\cdot) : G \rightarrow \mathbb{R}^m \).

• Let us consider the following problem:

\[ \hat{X} = \arg\min_{X \in G} \| \phi(X) \|^2_{\Sigma} \]

where \( \phi(\cdot) : G \to \mathbb{R}^m \).

• Classical iterative algorithm:

\[ X^{l+1} = X^l + \alpha^l \delta^{l+1/l} \]

where \( 0 < \alpha^l \leq 1 \) is a step size.
Let us consider the following problem:

\[ \hat{X} = \arg\min_{X \in G} \| \phi(X) \|^2_{\Sigma} \]

where \( \phi(\cdot) : G \to \mathbb{R}^m \).

Intrinsic iterative algorithm [1]:

\[ X^{l+1} = \exp_G^\wedge \left( \alpha^l \delta^{l+1/l} \right) X^l \]

where 0 < \( \alpha^l \leq 1 \) is a step size

Let us consider the following problem:

\[
\hat{X} = \arg\min_{X \in G} \| \phi(X) \|_2^2
\]

where \( \phi(\cdot): G \to \mathbb{R}^m \).

Intrinsic iterative algorithm [1]:

\[
X^{l+1} = \exp^\wedge_G (\alpha^l \delta^{l+1/l}) X^l
\]

where \( 0 < \alpha^l \leq 1 \) is a step size

\[
\delta^{l+1/l} = \arg\min_{\delta \in \mathbb{R}^p} \| \phi(X^l) + J_l \delta \|_\Sigma^2
\]

and \( J_l \) is the Lie group Jacobian of \( \phi \)

\[
J_l = \left. \frac{d\phi(\exp^\wedge_G(s) X^l)}{ds} \right|_{s=0}
\]

Let us consider the following problem:

\[ \hat{X} = \arg\min_{X \in G} \| \phi (X) \|_{\Sigma}^2 \]

where \( \phi (\cdot) : G \to \mathbb{R}^m \).

Intrinsic iterative algorithm [1]:

\[ X^{l+1} = \exp_G^\wedge (\alpha^l \delta^{l+1/l}) X^l \]

where \( 0 < \alpha^l \leq 1 \) is a step size

\[ \delta^{l+1/l} = \arg\min_{\delta \in \mathbb{R}^p} \| \phi (X^l) + J_l \delta \|_{\Sigma}^2 = - (J_l^T \Sigma^{-1} J_l)^{-1} J_l^T \Sigma^{-1} \phi (X^l) \]

and \( J_l \) is the Lie group Jacobian of \( \phi \)

\[ J_l = \frac{d\phi \left( \exp_G^\wedge (s) X^l \right)}{ds} \bigg|_{s=0} \]

Part 1: Outline

1. Introduction
2. Preliminaries on matrix Lie groups
3. Estimation tools (probability distribution, optimization,...)
4. Iterated Extended Kalman Filter on Lie Groups
5. Application to localization
Proposed Dynamic Model

State Model

• $X_k \in G \Rightarrow \text{state to estimate at time } k$

$$X_k = \exp^\wedge_G (n_k) f (X_{k-1}) \rightarrow p (X_k | X_{k-1}) = \mathcal{N}_G^R (X_k; f (X_{k-1}), Q_k)$$

• Example: $C_{k+1} = \exp_{SE(3)}^\wedge (v_k \Delta t) C_k$ and $v_{k+1} = v_k + n_k$
Proposed Dynamic Model

State Model

- \( X_k \in G \Rightarrow \) state to estimate at time \( k \)

\[
X_k = \exp^G(n_k) f(X_{k-1}) \rightarrow p(X_k|X_{k-1}) = \mathcal{N}_G^R(X_k; f(X_{k-1}), Q_k)
\]

- Example: \( C_{k+1} = \exp^\text{SE}(3) (v_k \Delta t) C_k \) and \( v_{k+1} = v_k + n_k \)

Observation Model

- \( Z_k \in G' \Rightarrow \) measurement available at time \( k \)

\[
Z_k = \exp^G(w_k) h(X_k) \rightarrow p(Z_k|X_k) = \mathcal{N}_{G'}^R(Z_k; h(X_k), R_k)
\]

- Example: \( Z_k = \exp^\text{SE}(3) (w_k) C_k \)
Objective

recursively approximate the posterior distribution with a CGD

\[ p(X_k|Z_1, \ldots, Z_k) \approx \mathcal{N}_G^R(X_k; \mu_{k|k}, P_{k|k}) \]
Objective

recursively approximate the posterior distribution with a CGD

\[ p(X_k|Z_1, \ldots, Z_k) \approx \mathcal{N}_G^R \left( X_k; \mu_{k|k}, P_{k|k} \right) \]

Euclidean case: \( G = \mathbb{R}^p \) and \( G' = \mathbb{R}^{p'} \) \( \Rightarrow \) CGD reduces to Mult. normal dist.
Objective

recursively approximate the posterior distribution with a CGD

\[ p(X_k|Z_1, \ldots, Z_k) \approx \mathcal{N}_G^{R}(X_k; \mu_{k|k}, P_{k|k}) \]

Euclidean case: \( G = \mathbb{R}^p \) and \( G' = \mathbb{R}^{p'} \) \( \Rightarrow \) CGD reduces to Mult. normal dist.

\[
\begin{align*}
\mu_{k-1|k-1}, P_{k-1|k-1} \\
\mu_{k|k-1}, P_{k|k-1} \\
\mu_{k|k}, P_{k|k} \\
\end{align*}
\]
Propagation Step

CGD hypotheses

\[ p (X_{k-1} | Z_1, \ldots, Z_{k-1}) = \mathcal{N}_G^R (X_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1}) \]

\[ p (X_k | X_{k-1}) = \mathcal{N}_G^R (X_k; f (X_{k-1}), Q_k) \]
Propagation Step

**CGD hypotheses**

\[
p (X_{k-1} | Z_1, \ldots, Z_{k-1}) = \mathcal{N}_G^R \left( X_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1} \right) \\
p (X_k | X_{k-1}) = \mathcal{N}_G^R \left( X_k; f (X_{k-1}), Q_k \right)
\]

- We want to compute (using Markov property):

\[
p (X_k | Z_1, \ldots, Z_{k-1}) = \int p (X_k | X_{k-1}) p (X_{k-1} | Z_1, \ldots, Z_{k-1}) \, d_H X_{k-1}
\]
We want to compute (using Markov property):

\[
p (X_k | Z_1, \ldots, Z_{k-1}) = \int p (X_k | X_{k-1}) p (X_{k-1} | Z_1, \ldots, Z_{k-1}) \, d_H X_{k-1}
\]

\[
\int \text{CGD} \times \text{CGD} \neq \text{CGD} !
\]
Propagation Step

**CGD hypotheses**

\[
p(X_{k-1}|Z_1, \ldots, Z_{k-1}) = \mathcal{N}_G^R (X_{k-1}; \mu_{k-1|k-1}, P_{k-1|k-1})
\]

\[
p(X_k|X_{k-1}) = \mathcal{N}_G^R (X_k; f (X_{k-1}), Q_k)
\]

- We want to compute (using Markov property):

\[
p(X_k|Z_1, \ldots, Z_{k-1}) = \int p(X_k|X_{k-1}) p(X_{k-1}|Z_1, \ldots, Z_{k-1}) d_{H}X_{k-1}
\]

\[
\int \text{CGD} \times \text{CGD} \neq \text{CGD} !
\]

\[
\Rightarrow \text{CGD fitting } p (X_k, X_{k-1}|Z_1, \ldots, Z_{k-1}) \approx \mathcal{N}_G^R \left( \begin{bmatrix} X_k \\ X_{k-1} \end{bmatrix}; \mu, P \right)
\]
Objective function

\[-\log (p(X_k | X_{k-1}) p(X_{k-1} | Z_1, \ldots, Z_{k-1}))\]

\[= \left\| \log_G (X_k f(X_{k-1})^{-1}) \right\|_{Q_k}^2 + \left\| \log_G \left( X_{k-1} (\mu_{k-1|k-1})^{-1} \right) \right\|_{P_{k-1|k-1}}^2\]
Propagation: Gauss-Laplace Approximation

Objective function

\[
-\log (p(X_k|X_{k-1}) p(X_{k-1}|Z_1, \ldots, Z_{k-1}))
= \left\| \log^V_G (X_k f (X_{k-1})^{-1}) \right\|^2_{Q_k} + \left\| \log^V_G (X_{k-1} (\mu_{k-1|k-1})^{-1}) \right\|^2_{P_{k-1|k-1}}
\]

- Fitted CGD

\[
p(X_k, X_{k-1}|Z_1, \ldots, Z_{k-1})
\approx \mathcal{N}^R_{G \times G} \left( \begin{bmatrix} X_k \\ X_{k-1} \end{bmatrix}; \begin{bmatrix} f (\mu_{k-1|k-1}) \\ \mu_{k-1|k-1} \end{bmatrix}, \begin{bmatrix} F_k P_{k-1|k-1} F_k^T + Q_k & F_k P_{k-1|k-1} \\ P_{k-1|k-1} F_k^T & P_{k-1|k-1} \end{bmatrix} \right)
\]

where \( F_k \) is the Lie group Jacobian of \( f \)
Objective function

\[
-\log (p(X_k|X_{k-1}) p(X_{k-1}|Z_1, \ldots, Z_{k-1}))
= \left\| \log^v_G (X_k f (X_{k-1})^{-1}) \right\|^2_{Q_k} + \left\| \log^v_G (X_{k-1} (\mu_{k-1|k-1})^{-1}) \right\|^2_{P_{k-1|k-1}}
\]

- Fitted CGD

\[
p(X_k, X_{k-1}|Z_1, \ldots, Z_{k-1}) \approx \mathcal{N}^R_G \left( \begin{bmatrix} X_k \\ X_{k-1} \end{bmatrix}; \begin{bmatrix} f(\mu_{k-1|k-1}) \\ \mu_{k-1|k-1} \end{bmatrix}, \begin{bmatrix} F_k P_{k-1|k-1} F^T_k + Q_k \\ P_{k-1|k-1} F_k^T \\ P_{k-1|k-1} F_k \end{bmatrix} \right)
\]

where $F_k$ is the Lie group Jacobian of $f$

- Marginalizing $X_{k-1}$ yields

\[
p(X_k|Z_1, \ldots, Z_{k-1}) \approx \mathcal{N}^R_G (X_k; \mu_{k|k-1} = f(\mu_{k-1|k-1}), P_{k|k-1} = F_k P_{k-1|k-1} F^T_k + Q_k)
\]

⇒ Using Lie group geodesics leads to classical propagation equations
Update Step

CGD hypotheses

\[
p(X_k|Z_1, \ldots, Z_{k-1}) = \mathcal{N}_G^R (X_k; \mu_{k|k-1}, P_{k|k-1})
\]

\[
p(Z_k|X_k) = \mathcal{N}_{G'}^R (Z_k; h(X_k), R_k)
\]
Update Step

**CGD hypotheses**

\[
p(X_k|Z_1, \ldots, Z_{k-1}) = \mathcal{N}^R_G \left( X_k; \mu_{k|k-1}, P_{k|k-1} \right)
\]

\[
p(Z_k|X_k) = \mathcal{N}^R_{G'} \left( Z_k; h(X_k), R_k \right)
\]

- We want to compute (using Markov property):

\[
p(X_k|Z_1, \ldots, Z_k) \propto p(Z_k|X_k) p(X_k|Z_1, \ldots, Z_{k-1})
\]
We want to compute (using Markov property):

\[ p(X_k|Z_1,\ldots,Z_{k-1}) \propto p(Z_k|X_k) p(X_k|Z_1,\ldots,Z_{k-1}) \]

\[ CGD \times CGD \neq CGD ! \]
Update Step

CGD hypotheses

\[ p(X_k|Z_1, \ldots, Z_{k-1}) = \mathcal{N}_G^R \left( X_k; \mu_{k|k-1}, P_{k|k-1} \right) \]
\[ p(Z_k|X_k) = \mathcal{N}_{G'}^R \left( Z_k; h(X_k), R_k \right) \]

- We want to compute (using Markov property):

\[ p(X_k|Z_1, \ldots, Z_k) \propto p(Z_k|X_k) p(X_k|Z_1, \ldots, Z_{k-1}) \]

CGD $\times$ CGD $\neq$ CGD!

$\Rightarrow$ CGD fitting
\[ p(X_k|Z_1, \ldots, Z_k) \approx \mathcal{N}_G^R \left( X_k; \mu_{k|k}, P_{k|k} \right) \]
Propagation: Gauss-Laplace approximation

Objective function

\[-\log (p(Z_k|X_k) p(X_k|Z_1, \ldots, Z_{k-1})) \]
\[= \left\| \log_G^\vee (Z_k h(X_k)^{-1}) \right\|^2_{R_k} + \left\| \log_G^\vee (X_k (\mu_{k|k-1})^{-1}) \right\|^2_{P_{k|k-1}} \]
Propagation: Gauss-Laplace approximation

Objective function

\[-\log(p(Z_k|X_k) p(X_k|Z_1, \ldots, Z_{k-1})) \]

\[= \left\| \log_{G'} (Z_k h(X_k)^{-1}) \right\|^2_{R_k} + \left\| \log_{G} (X_k (\mu_{k|k-1})^{-1}) \right\|^2_{P_{k|k-1}} \]

Finding the mode \(\Rightarrow\) Gauss-Newton on Lie groups
Propagation: Gauss-Laplace approximation

Intrinsic Gauss-Newton iteration:

\[ X^{l+1} = \exp_G^\wedge (\delta^{l+1/l}) \ X^l \approx \exp_G^\wedge \left( K_l \left( \log_{G'}^\vee \left( Z_{k \ h \ (X^l)}^{-1} \right) + J_l \delta^l \right) \right) \mu_{k|k-1} \]

- \( \delta^l = \log_G^\vee \left( X^l \mu_{k|k-1}^{-1} \right) \)
- \( J_l \) Lie group Jacobian of \( h \)
- \( K_l \) Lie-Kalman gain (\( \Phi_l \equiv \Phi_G (\delta^l) \))

\[ K_l = P_{k|k-1} \Phi_l^T J_l^T \left( J_l \Phi_l P_{k|k-1} \Phi_l^T J_l^T + R_k \right)^{-1} \]
Intrinsic Gauss-Newton iteration:

\[ X^{l+1} = \exp_G^\wedge \left( \delta^{l+1/l} \right) X^l \approx \exp_G^\wedge \left( K_l \left( \log_G \left( Z_k h \left( X^l \right)^{-1} \right) + J_l \delta^l \right) \right) \mu_{k|k-1} \]

- \( \delta^l = \log_G \left( X^l \mu_{k|k-1}^{-1} \right) \)
- \( J_l \) Lie group Jacobian of \( h \)
- \( K_l \) Lie-Kalman gain (\( \Phi_l \equiv \Phi_G \left( \delta^l \right) \))

\[ K_l = P_{k|k-1} \Phi_l^T J_l^T \left( J_l \Phi_l P_{k|k-1} \Phi_l^T J_l^T + R_k \right)^{-1} \]

At convergence, we take \( \mu_{k|k} = X^l \).
Propagation: Gauss-Laplace approximation

Intrinsic Gauss-Newton iteration:

\[
X^{l+1} = \exp_G^\wedge (\delta^{l+1/l}) \ X^l \approx \exp_G^\wedge \left( K_l \left( \log_G^\vee \left( Z_l h (X^l)^{-1} \right) + J_l \delta^l \right) \right) \mu_{k|k-1}
\]

- \( \delta^l = \log_G^\vee \left( X^l \mu_{k|k-1}^{-1} \right) \)
- \( J_l \) Lie group Jacobian of \( h \)
- \( K_l \) Lie-Kalman gain (\( \Phi_l \equiv \Phi_G (\delta^l) \))

\[
K_l = P_{k|k-1} \Phi_l^T J_l^T \left( J_l \Phi_l P_{k|k-1} \Phi_l^T J_l^T + R_k \right)^{-1}
\]

At convergence, we take \( \mu_{k|k} = X^l \).

Fitted CGD

\[
p(X_k|Z_1, \ldots, Z_k) \approx N_G^R \left( X_k; \mu_{k|k}, P_{k|k} = \Phi_l \left( \text{Id} - K_l J_l \Phi_l \right) P_{k|k-1} \Phi_l^T \right)
\]

\( \Rightarrow \text{Using Lie group geodesics leads to the generalization of IEKF update equations} \)
## LG-IEKF Algorithm

<table>
<thead>
<tr>
<th>IEKF on Euclidean spaces</th>
<th>IEKF on matrix Lie groups</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs:</strong> $\mu_{k-1</td>
<td>k-1}$, $P_{k-1</td>
</tr>
<tr>
<td><strong>Outputs:</strong> $\mu_{k</td>
<td>k}$, $P_{k</td>
</tr>
<tr>
<td>1) Propagation</td>
<td>1) Propagation</td>
</tr>
<tr>
<td>$\mu_{k</td>
<td>k-1} = f\left(\mu_{k-1</td>
</tr>
<tr>
<td>$P_{k</td>
<td>k-1} = Q_k + F_k P_{k-1</td>
</tr>
<tr>
<td>2) Update</td>
<td>2) Update</td>
</tr>
<tr>
<td>Set $X^0 = \mu_{k</td>
<td>k-1}$</td>
</tr>
<tr>
<td>Iterate until convergence:</td>
<td>Iterate until convergence:</td>
</tr>
<tr>
<td>$K_l = P_{k</td>
<td>k-1} J_l^T \left( J_l P_{k</td>
</tr>
<tr>
<td>$\delta^{l+1} = K_l \left( Z_k - h\left(X^l\right) + J_l \delta^l \right)$</td>
<td>$\delta^{l+1} = K_l \left( \log^\vee_{G'} \left(Z_k h\left(X^l\right)^{-1}\right) + J_l \delta^l \right)$</td>
</tr>
<tr>
<td>$X^{l+1} = \delta^{l+1} + \mu_{k</td>
<td>k-1}$</td>
</tr>
<tr>
<td>At convergence:</td>
<td>At convergence:</td>
</tr>
<tr>
<td>$\mu_{k</td>
<td>k} = X^l$</td>
</tr>
<tr>
<td>$P_{k</td>
<td>k} = (I_d - K_l J_l) P_{k</td>
</tr>
</tbody>
</table>
Part 1: Outline

1 Introduction
2 Preliminaries on matrix Lie groups
3 Estimation tools (probability distribution, optimization,...)
4 Iterated Extended Kalman Filter on Lie Groups
5 Application to localization
Context

Video frame examples

Annotated Database
Proposed solution

- **Objective** ⇒ improve CBIR output
Proposed solution

- **Objective** → improve CBIR output
- **Formulation of a target tracking problem on** $SE(3)$
Proposed solution

- **Objective** ⇒ improve CBIR output

- **Formulation of a target tracking problem on** $SE(3)$

- **Rao-Blackwellized particle smoother on Lie groups (LG-RBPS)** which involves LG-IEKF and LG-RTS
  - measurements are map coordinates from CBIR
  - **Virtual Measurements** to prevent the particles from crossing walls
Problem formulation

- The camera should have a smooth trajectory

\[ C_{k+1} = \exp^{\wedge}_{SE(3)} (v_k \Delta t) C_k \quad \text{and} \quad v_{k+1} = v_k + n_k \]
Problem formulation

- The camera should have a smooth trajectory
  \[ C_{k+1} = \exp_{SE(3)}^\wedge (v_k \Delta t) \ C_k \quad \text{and} \quad v_{k+1} = v_k + n_k \]

- Measurement model
  \[ Z_k = \begin{bmatrix} Z_k(1), Z_k(2), \ldots, Z_k(N_{VM}), & Z_k(N_{VM} + 1), \ldots, Z_k(N_{VM} + N_{CBIR}) \end{bmatrix} \]

Virtual Measurements  
CBIR output
Problem formulation

- The camera should have a smooth trajectory
  \[ C_{k+1} = \exp_{SE(3)}^{\wedge} (v_k \Delta t) C_k \quad \text{and} \quad v_{k+1} = v_k + n_k \]

- Measurement model

\[ Z_k = \left[ Z_k(1), Z_k(2), \ldots, Z_k(N_{VM}), Z_k(N_{VM} + 1), \ldots, Z_k(N_{VM} + N_{CBIR}) \right] \]

- Virtual Measurements
- CBIR output

\( s_k \) discrete selection variable
Problem formulation

- The camera should have a smooth trajectory
  \[ C_{k+1} = \exp_{SE(3)}^\wedge (v_k \Delta t) C_k \quad \text{and} \quad v_{k+1} = v_k + n_k \]

- Measurement model
  \[ Z_k = \begin{bmatrix} Z_k(1), Z_k(2), \ldots, Z_k(N_{VM}), \underbrace{Z_k(N_{VM}+1), \ldots, Z_k(N_{VM}+N_{CBIR})}_{\text{Virtual Measurements}} \end{bmatrix} \]

  \( s_k \) discrete selection variable

  \[ p(Z_k|C_k, s_k = i) \]
  \[ = \prod_{j=1}^{N_{VM}+N_{CBIR}} p(Z_k(j)|C_k, s_k = i) \]
  \[ \propto \mathcal{N}_{SE(3)}^R (Z_k(i); C_k, Q_{k,i}) \]
Problem formulation

- The camera should have a smooth trajectory

\[ C_{k+1} = \exp_{SE(3)} (\nu_k \Delta t) \ C_k \quad \text{and} \quad \nu_{k+1} = \nu_k + n_k \]

- Measurement model

\[
Z_k = \begin{bmatrix}
Z_k(1), Z_k(2), ..., Z_k(N_{VM}), & Z_k(N_{VM} + 1), ..., & Z_k(N_{VM} + N_{CBIR})
\end{bmatrix}
\]

Virtual Measurements \hspace{1cm} CBIR output

\( s_k \) discrete selection variable

\[
p(Z_k|C_k, s_k = i) = \prod_{j=1}^{N_{VM}+N_{CBIR}} p(Z_k(j)|C_k, s_k = i) \propto \mathcal{N}_{SE(3)}(Z_k(i); C_k, Q_k, i)
\]

- \( p(s_k|s_{k-1}) = p(s_k), p(s_k = i) \ll p(s_k = j) \) for \( i \leq N_{VM} \) and \( j > N_{VM} \)
Problem formulation

- The camera should have a smooth trajectory
  \[ C_{k+1} = \exp_{SE(3)}^\wedge (v_k \Delta t) C_k \] and \( v_{k+1} = v_k + n_k \)

- Measurement model
  \[
  Z_k = \begin{bmatrix}
  Z_k(1), Z_k(2), \ldots, Z_k(N_{VM}), Z_k(N_{VM} + 1), \ldots, Z_k(N_{VM} + N_{CBIR})
  \end{bmatrix}
  \]
  Virtual Measurements
  CBIR output

- \( s_k \) discrete selection variable
  \[
  p(Z_k|C_k, s_k = i) = \prod_{j=1}^{N_{VM} + N_{CBIR}} p(Z_k(j)|C_k, s_k = i)
  \]
  \[
  \propto \mathcal{N}_{SE(3)}^R (Z_k(i); C_k, Q_k, i)
  \]

- \( p(s_k|s_{k-1}) = p(s_k), p(s_k = i) \ll p(s_k = j) \) for \( i \leq N_{VM} \) and \( j > N_{VM} \)
- LG-RBPS [1] to solve the problem (\( s_k \) sampled, \( X_k \) marginalized)

### Results

<table>
<thead>
<tr>
<th>Method</th>
<th>GO80</th>
<th>GO81</th>
<th>GO82</th>
<th>GO83</th>
<th>GO84</th>
<th>GO85</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBIR only</td>
<td>1.7</td>
<td>1.3</td>
<td>2.4</td>
<td>2.2</td>
<td>1.3</td>
<td>2.2</td>
</tr>
<tr>
<td>CBIR + LG-RBPS No VM.</td>
<td>0.5</td>
<td>0.7</td>
<td>1.7</td>
<td>0.9</td>
<td>&lt;0.5</td>
<td>1.3</td>
</tr>
<tr>
<td>CBIR + LG-RBPS With VM.</td>
<td>0.5</td>
<td>&lt;0.5</td>
<td>0.8</td>
<td>0.7</td>
<td>&lt;0.5</td>
<td>0.9</td>
</tr>
</tbody>
</table>

RMSE in meters of the estimated trajectories
### Results

<table>
<thead>
<tr>
<th>Method</th>
<th>GO80</th>
<th>GO81</th>
<th>GO82</th>
<th>GO83</th>
<th>GO84</th>
<th>GO85</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBIR only</td>
<td>1.7</td>
<td>1.3</td>
<td>2.4</td>
<td>2.2</td>
<td>1.3</td>
<td>2.2</td>
</tr>
<tr>
<td>CBIR + LG-RBPS No VM.</td>
<td>0.5</td>
<td>0.7</td>
<td>1.7</td>
<td>0.9</td>
<td>&lt;0.5</td>
<td>1.3</td>
</tr>
<tr>
<td>CBIR + LG-RBPS With VM.</td>
<td>0.5</td>
<td>&lt;0.5</td>
<td>0.8</td>
<td>0.7</td>
<td>&lt;0.5</td>
<td>0.9</td>
</tr>
</tbody>
</table>

RMSE in meters of the estimated trajectories

---

**Results on GO82**

- **CBIR (NN)**
- **CBIR+LG-RBPS no VM**
- **CBIR+LG-RBPS with VM**
- **Ground truth**
Results

Movie
Part II

Structure from Motion for ordered images
How to build automatically an annotated database?
How to build automatically an annotated database?

1. Take a training video
How to build automatically an annotated database?

1. Take a training video
2. Apply a Structure from Motion (SfM) algorithm
How to build automatically an annotated database?

1. Take a training video
2. Apply a Structure from Motion (SfM) algorithm
3. Align the estimated trajectory with the 2D plan (by hand)
Structure from Motion: State of the art

<table>
<thead>
<tr>
<th></th>
<th>SfM for unordered images (ex: [1])</th>
<th>Monocular Visual SLAM (ex: [2])</th>
<th>Objective: SfM for ordered images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Comp. time</td>
<td>-</td>
<td>++</td>
<td>+</td>
</tr>
</tbody>
</table>

### Structure from Motion: State of the art

<table>
<thead>
<tr>
<th></th>
<th>SfM for unordered images (ex: [1])</th>
<th>Monocular Visual SLAM (ex: [2])</th>
<th>Objective: SfM for ordered images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness</td>
<td>+</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>Comp. time</td>
<td>−</td>
<td>++</td>
<td>+</td>
</tr>
</tbody>
</table>

How to combine the best of both worlds?

---

Part 2: Outline

1. Introduction
2. System Architecture
3. Relative Similarity Denoising
4. Submap Reconstruction
5. Results
Part 2: Outline

1. Introduction

2. System Architecture

3. Relative Similarity Denoising

4. Submap Reconstruction

5. Results
A monocular Visual SLAM system architecture

Keyframe Selection

video

keyframes, tracks

Intrinsic parameter estimation on Lie groups: Applications in mapping and localization from a wearable camera
A monocular Visual SLAM system architecture

Keyframe Selection ➔ Submap Reconstruction ➔ ... ➔ Submap Reconstruction ➔ Submap Reconstruction

video

keyframes, tracks

submaps (point clouds and camera poses + covariances from CGD fitting)
A monocular Visual SLAM system architecture

- **Keyframe Selection**
- **Submap Reconstruction**
  - **Pairwise Similarity Estimation**
    - **Relative Similarities** between submaps
      - Relative similarities between submaps + covariances from CGD fitting
    - **Submap Reconstruction**
      - **Submap Reconstruction**
        - **Submap Reconstruction**
          - **Submap Reconstruction**

Intrinsic parameter estimation on Lie groups: Applications in mapping and localization from a wearable camera
A monocular Visual SLAM system architecture

Keyframe Selection → Submap Reconstruction → Submap Reconstruction → Submap Reconstruction

video

keyframes, tracks

aligned submaps

submaps (point clouds and camera poses + covariances from CGD fitting)

Pairwise Similarity Estimation

relative similarities between submaps + covariances from CGD fitting

Relative Similarity Denoising

How to add robustness in submap reconstruction and relative similarity denoising modules?
How to add robustness in submap reconstruction and relative similarity denoising modules?

Intrinsic parameter estimation on Lie groups: Applications in mapping and localization from a wearable camera
Part 2: Outline

1 Introduction

2 System Architecture

3 Relative Similarity Denoising

4 Submap Reconstruction

5 Results
From the pairwise similarity estimation module:

\( \mathcal{N}_{\text{Sim}(3)}^R (Z_{ij}; \bar{Z}_{ij}, \Sigma_{ij}) \)

where \( Z_{ij} \) is the similarity between submap \( j \) and submap \( i \)
Relative Similarity Denoising

From the pairwise similarity estimation module:

\[ \mathcal{N}^R_{\text{Sim}(3)} (Z_{ij}; \bar{Z}_{ij}, \Sigma_{ij}) \]

where \( Z_{ij} \) is the similarity between submap \( j \) and submap \( i \)

**Objective**

Estimating the absolute similarities \( X_i^S \) from the relative similarities assuming

\[ Z_{ij} = \exp^\wedge_{\text{Sim}(3)} (\epsilon_{ij}^i) X_i^S X_j^{-1}_S \]

where \( \epsilon_{ij}^i \sim \mathcal{N}_{\mathbb{R}^7} (\epsilon_{ij}^i; 0_{7 \times 1}, \Sigma_{ij}^i) \) is a white Gaussian noise
Relative Similarity Denoising

From the pairwise similarity estimation module:

\[ \mathcal{N}_{Sim(3)}^{R}(Z_{ij}; \overline{Z}_{ij}, \Sigma_{ij}) \]

where \( Z_{ij} \) is the similarity between submap \( j \) and submap \( i \)

Objective

Estimating the absolute similarities \( X_{iS} \) from the relative similarities assuming

\[ Z_{ij} = \exp_{Sim(3)}^{\wedge}(\epsilon_{ij}) \ X_{iS} X_{jS}^{-1} \]

where \( \epsilon_{ij} \sim \mathcal{N}_{\mathbb{R}^7}(\epsilon_{ij}; 0_{7 \times 1}, \Sigma_{ij}) \) is a white Gaussian noise
Relative Similarity Denoising

From the pairwise similarity estimation module:

\[ \mathcal{N}^R_{Sim(3)} \left( Z_{ij}; \overline{Z}_{ij}, \Sigma_{ij} \right) \]

where \( Z_{ij} \) is the similarity between submap \( j \) and submap \( i \)

**Objective**

Estimating the absolute similarities \( X_{iS} \) from the relative similarities assuming

\[ Z_{ij} = \exp_{Sim(3)}^\wedge \left( \epsilon_{ij}^i \right) X_{iS} X_{jS}^{-1} \]

where \( \epsilon_{ij}^i \sim \mathcal{N}_{\mathbb{R}^7} \left( \epsilon_{ij}^i; 0_{7 \times 1}, \Sigma_{ij}^i \right) \) is a white Gaussian noise

1. **Remove outliers**
   - Assumption: static scene \( \Rightarrow Z_{(i+1)i} \) not outliers
   - For each \( Z_{ij} \) with \( (i \neq j + 1) \)
     - Find the shortest path of inliers between \( i \) and \( j \)
     - Compute the loop error \( \epsilon \) and its covariance \( P \)
     - Statistical test on \( Sim(3) \)
       \[ \epsilon^T P^{-1} \epsilon < t_{\chi^2} \Rightarrow Z_{ij} \text{ inlier} \]
Relative Similarity Denoising

From the pairwise similarity estimation module:

\[ \mathcal{N}_{Sim(3)}^R(Z_{ij}; \overline{Z}_{ij}, \Sigma_{ij}) \]

where \( Z_{ij} \) is the similarity between submap \( j \) and submap \( i \)

**Objective**

Estimating the absolute similarities \( X_{iS} \) from the relative similarities assuming

\[ Z_{ij} = \exp_{Sim(3)}^\wedge (\epsilon_{ij}^i) X_{iS} X_{jS}^{-1} \]

where \( \epsilon_{ij}^i \sim \mathcal{N}_{\mathbb{R}^7}(\epsilon_{ij}^i; 0_{7 \times 1}, \Sigma_{ij}^i) \) is a white Gaussian noise

1. Remove outliers
   - Assumption: static scene \( \Rightarrow Z_{(i+1)i} \) not outliers
   - For each \( Z_{ij} \) with \( (i \neq j + 1) \)
     - Find the shortest path of inliers between \( i \) and \( j \)
     - Compute the loop error \( \epsilon \) and its covariance \( P \)
     - Statistical test on \( Sim(3) \)
       \[ \epsilon^T P^{-1} \epsilon < t_{\chi^2} \Rightarrow Z_{ij} \text{ inlier} \]

2. Estimate absolute similarities
   \[
   \arg\min_{\{X_{iS}\}_{i \in \mathcal{V}}} \left( \sum_{i,j} \left\| \log_{Sim(3)}^\vee (Z_{ij} X_{jS} X_{iS}^{-1}) \right\|_2^2 \right)
   \]
Part 2: Outline

1. Introduction
2. System Architecture
3. Relative Similarity Denoising
4. Submap Reconstruction
5. Results
Submap Reconstruction

Based on the *Known Rotation Problem* [1] (SfM for unordered images)

Based on the **Known Rotation Problem** [1] (SfM for unordered images)

Submap Reconstruction

Based on the *Known Rotation Problem* [1] (SfM for unordered images)

Submap Reconstruction

Based on the **Known Rotation Problem** [1] (SfM for unordered images)

Based on the *Known Rotation Problem* [1] (SfM for unordered images)

Part 2: Outline

1. Introduction
2. System Architecture
3. Relative Similarity Denoising
4. Submap Reconstruction
5. Results
## Results: TUM dataset

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>fr2/desk</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>2.22</td>
<td>4.52</td>
<td>13.50</td>
<td>×</td>
<td>1.77</td>
<td>9.5</td>
</tr>
<tr>
<td>fr2/xyz</td>
<td>1.28</td>
<td>1.47</td>
<td>3.79</td>
<td>24.28</td>
<td>1.18</td>
<td>2.6</td>
</tr>
</tbody>
</table>

RMSE (cm) of the estimated trajectories

---

Lim et al., “Real-time 6-dof monocular VSLAM in a large-scale environment”, ICRA 2014
Results: “apartment” dataset

- 20 min video
- 6000 selected keyframes
- 5h processing (Matlab coded)
Conclusion

- Estimation tools on Lie groups
  - Concentrated Gaussian distribution on Lie groups
  - Intrinsic Gauss-Newton algorithm on Lie groups
  - Generalization of IEKF (and RTS) to Lie groups using intrinsic Gauss-Laplace approximation

- Application of LG-IEKF and LG-RTS through LG-RBPS in a Visual Indoor Localization problem

- New SfM for ordered images approach to build automatically the database
Conclusion

• Estimation tools on Lie groups
  • Concentrated Gaussian distribution on Lie groups
  • Intrinsic Gauss-Newton algorithm on Lie groups
  • Generalization of IEKF (and RTS) to Lie groups using intrinsic Gauss-Laplace approximation

• Application of LG-IEKF and LG-RTS through LG-RBPS in a Visual Indoor Localization problem
Conclusion

- Estimation tools on Lie groups
  - Concentrated Gaussian distribution on Lie groups
  - Intrinsic Gauss-Newton algorithm on Lie groups
  - Generalization of IEKF (and RTS) to Lie groups using intrinsic Gauss-Laplace approximation

- Application of LG-IEKF and LG-RTS through LG-RBPS in a Visual Indoor Localization problem

- New SfM for ordered images approach to build automatically the database
Future Work

- Filtering/Smoothing on Lie groups
  - CGD normalization constant $\alpha(X) \neq \frac{1}{\sqrt{(2\pi)^p \det(P)}}$
Future Work

- Filtering/Smoothing on Lie groups
  - CGD normalization constant $\alpha(X) \neq \frac{1}{\sqrt{(2\pi)^p \det(P)}}$

- Visual Indoor Localization
  - Include robust visual odometry/VSLAM to help in the guidance of the particles
Future Work

- Filtering/Smoothing on Lie groups
  - CGD normalization constant $\alpha (X) \neq \frac{1}{\sqrt{(2\pi)^p \det(P)}}$
- Visual Indoor Localization
  - Include robust visual odometry/VSLAM to help in the guidance of the particles

Papers ⇒ https://sites.google.com/site/guillaumebourmaud/
Thank you for your attention