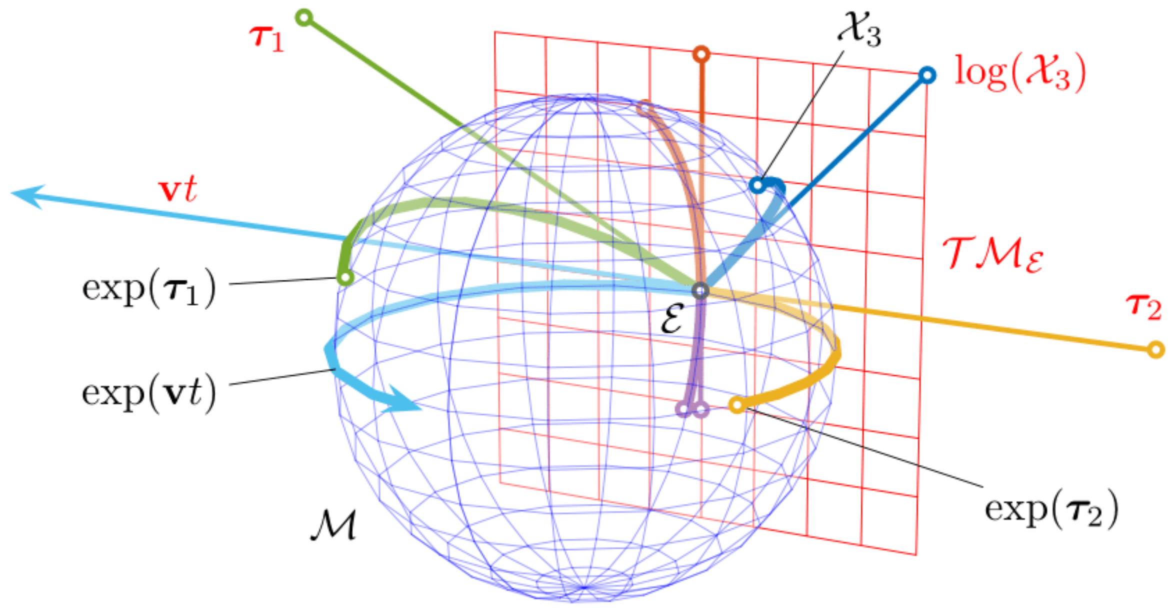
EECE 5550: Mobile Robotics



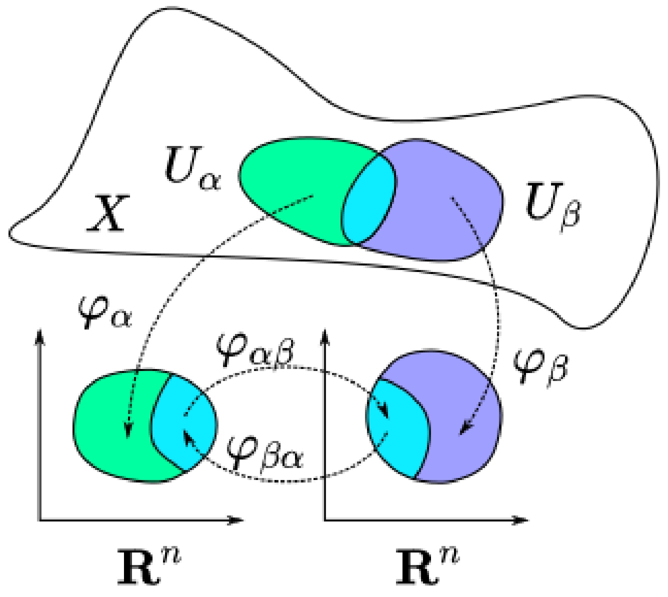
Lecture 3: An Introduction to Manifolds and Lie Groups (Part II)

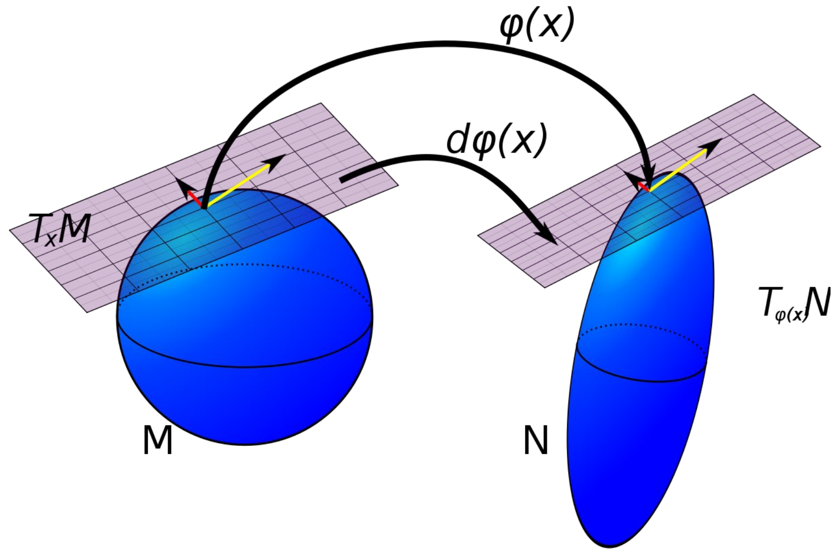
Recap

Last time: Smooth manifolds

- Basic definitions (charts, coordinates, etc.)
- Calculus on manifolds
- Diffeomorphisms, submersions and their *local* counterparts
- The Inverse Function and Preimage Theorems

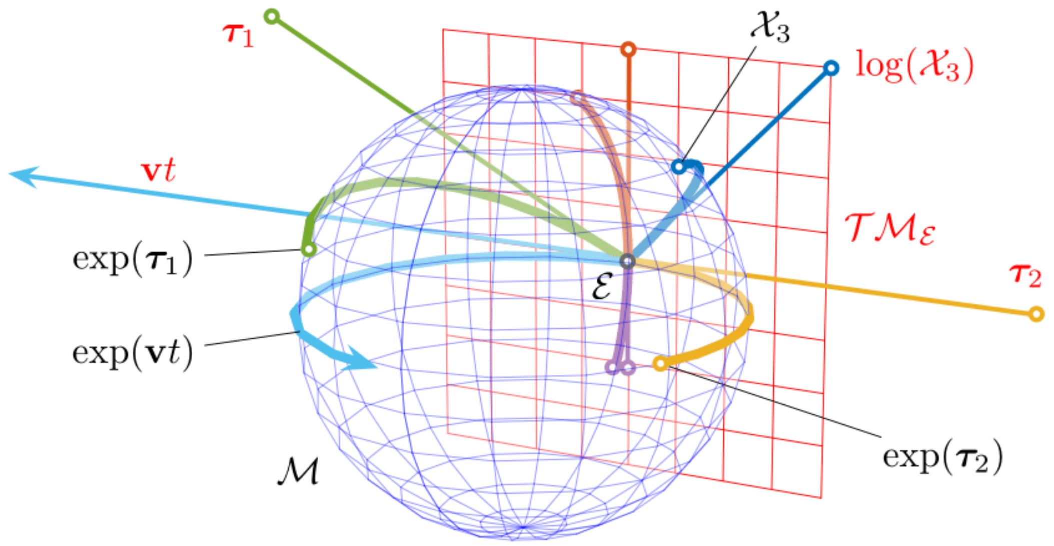
Today: Groups and Lie groups



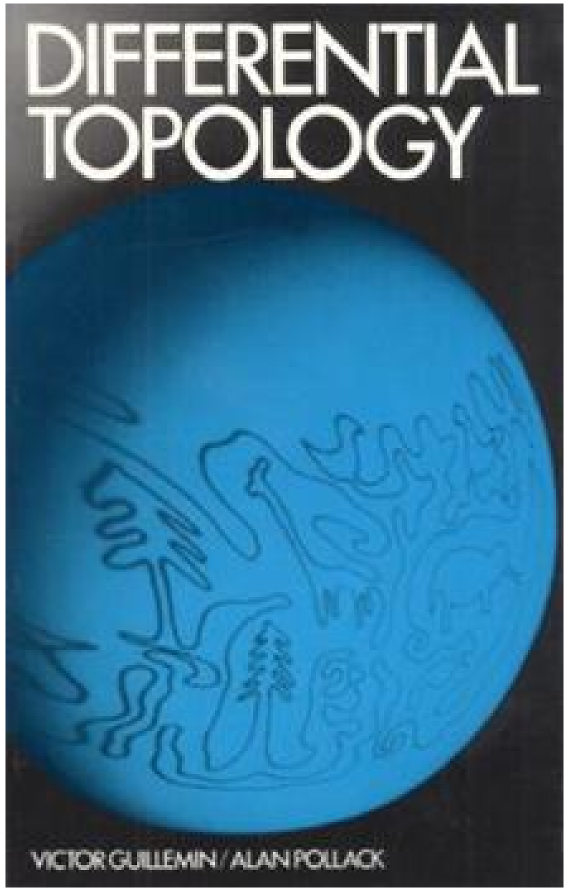


Plan of the day

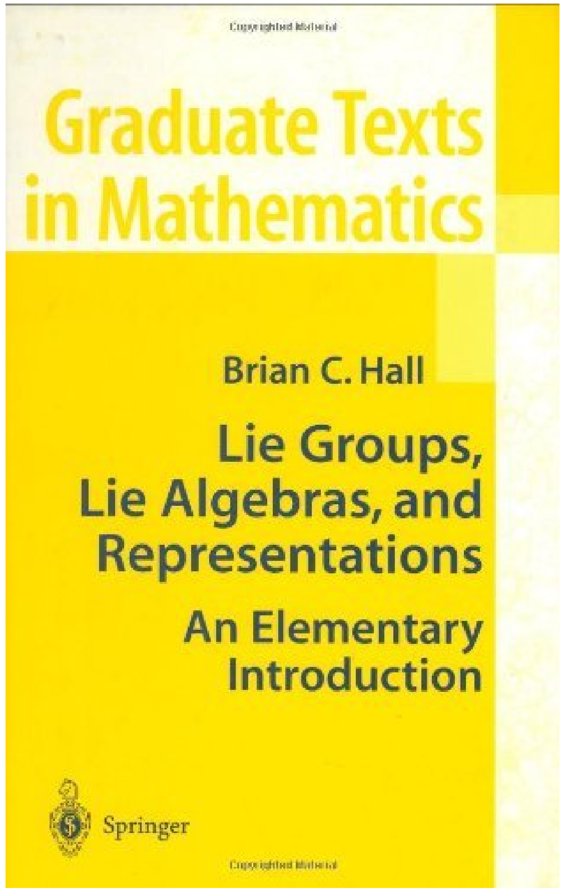
- Groups and group actions
- Lie groups



References



Differential Topology
(Guillemin and Pollack)


Copyrighted Material

Graduate Texts in Mathematics

Brian C. Hall

Lie Groups, Lie Algebras, and Representations

An Elementary Introduction

Springer

Copyrighted Material

Lie Groups, Lie Algebras,
and Representations (Hall)

Papers:

- J. Sola, J. Deray, D. Atchuthan: “A Micro Lie Theory for State Estimation in Robotics”
- E. Eade: “Lie Groups for 2D and 3D Transformations”

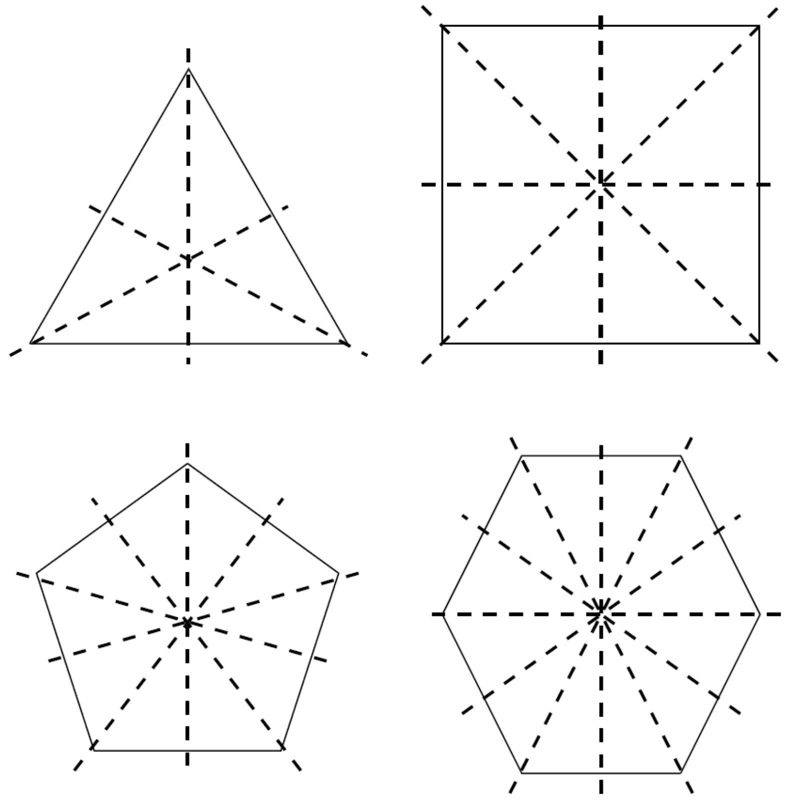
Groups

A **group** is a set G together with a binary operation $\cdot$ that satisfies the following three *group axioms*:

- **Associativity:** $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ for all $x, y, z \in G$
- **Identity:** There is an *identity element* $e \in G$ such that $e \cdot g = g \cdot e = g$ for all $g \in G$.
- **Invertibility:** Every element $g \in G$ has an *inverse* in G, denoted g^{-1}, satisfying $gg^{-1} = g^{-1}g = e$.

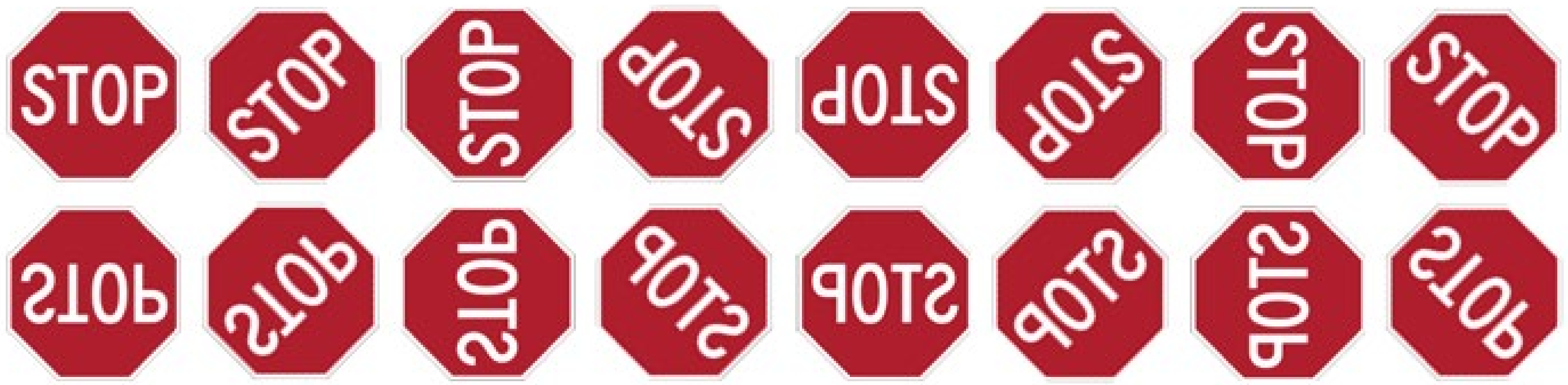
Key points:

- Super simple to define, but an *extremely rich* theory
- **LOTS** of cool examples ☺!
  - Integers under addition
  - Symmetry groups of regular polygons
  - Invertible maps $f: X \to X$ under composition


Symmetries of regular polygons

Example: Dihedral groups

The symmetry group of a regular polygon is called a *dihedral* group



Symmetries of the octagon

- The symmetry group of the n-gon is denoted by D_{2n}
- It has $2n$ elements, generated by *rotations* of $2\pi / n$ and *reflection*

Group actions

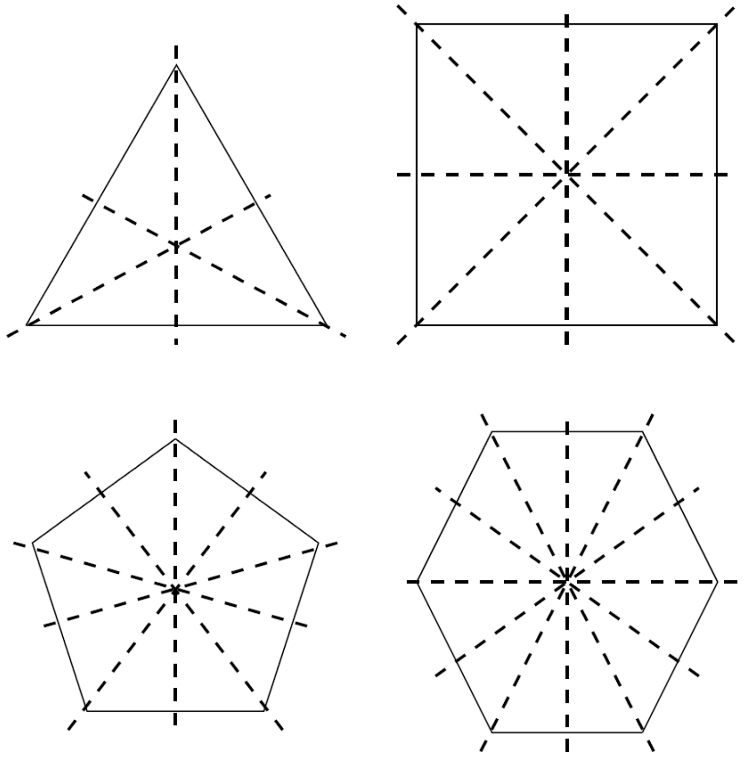
Let *G* be a group with identity *e* and *X* a set. A *left-action* of *G* on *X* is a binary map $\star: G \times X \to X$ satisfying the following two axioms:

- **Identity:** $e \star x = x$ for all *x* ∈ *X.*
- **Compatibility:** $g \star (h \star x) = gh \star x$ for all *g, h* ∈ *G* and *x* ∈ *X.*

Key points:

- Actions are one of the most important concepts in group theory (Historical note: Group theory originated in *geometry* as the study of *symmetries*)

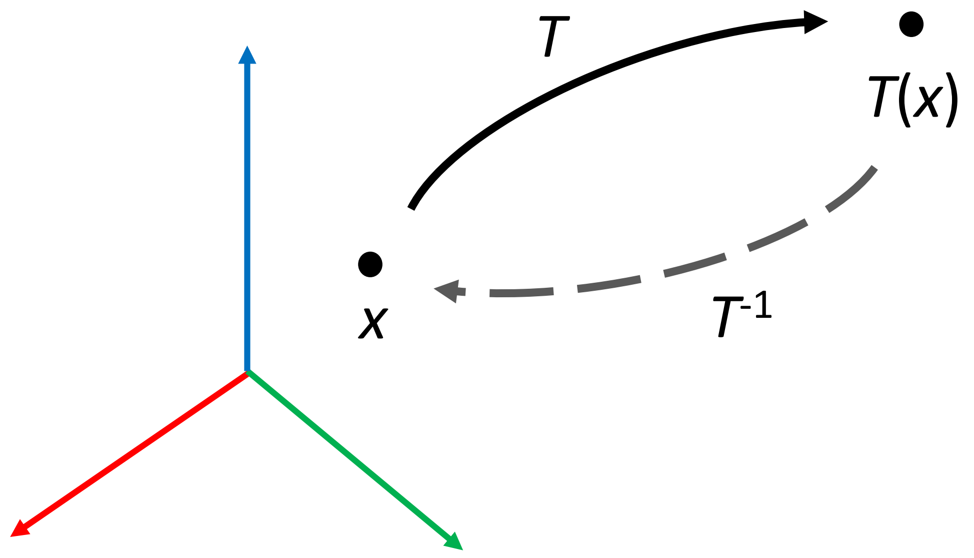
- **Examples:**
  - Dihedral groups act on vertices of regular polygons
  - The affine group $Aff(n)$ acts on $\mathbb{R}^n$!


Symmetries of regular polygons

Example: Coordinate transformations act on $\mathbb{R}^n$

$$\begin{tikzcd}
\text{Linear} & \text{Affine} \\
GL(d) \arrow[r] & Aff(d) \\
O(d) \arrow[r] \arrow[u] & E(d) \arrow[u] \\
SO(d) \arrow[r] \arrow[u] & SE(d) \arrow[u]
\end{tikzcd}$$

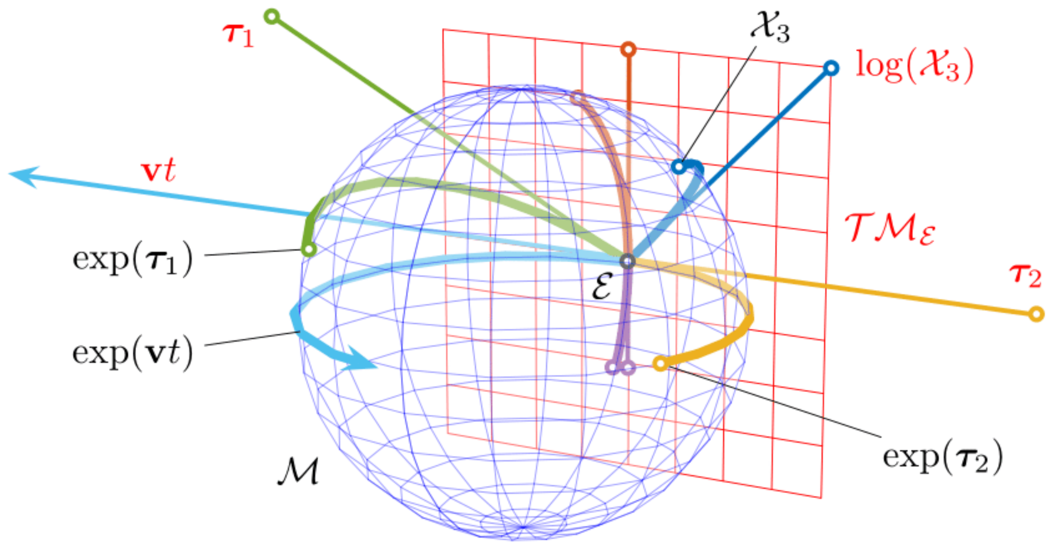
Subgroups of the affine group Aff(n)



$$T(x) = Ax + b,$$
$$A \in GL(n), \qquad b \in \mathbb{R}^n$$

Plan of the day

- Groups and group actions
- Lie groups



Lie Groups

A *Lie group* G is a *group* whose elements *also* form a smooth *manifold*, such that multiplication and inversion are smooth maps.

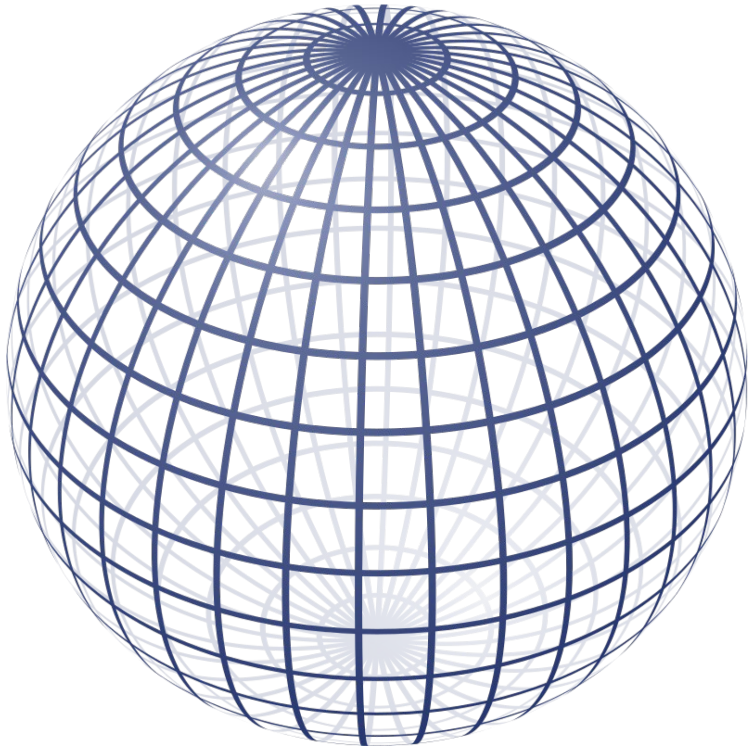
Intuition: A Lie group is a "continuous" group
(In fact, Lie groups were originally studied as *continuous transformation groups* in geometry)

Key points:

- Compatibility of *algebraic* and *smooth* structures is an *extremely strong* condition
  ⇒ Lie groups are "highly regular" objects
  ⇒ **LOTS** of things we can say about them that don't hold for general manifolds

- They are super pretty! 😍

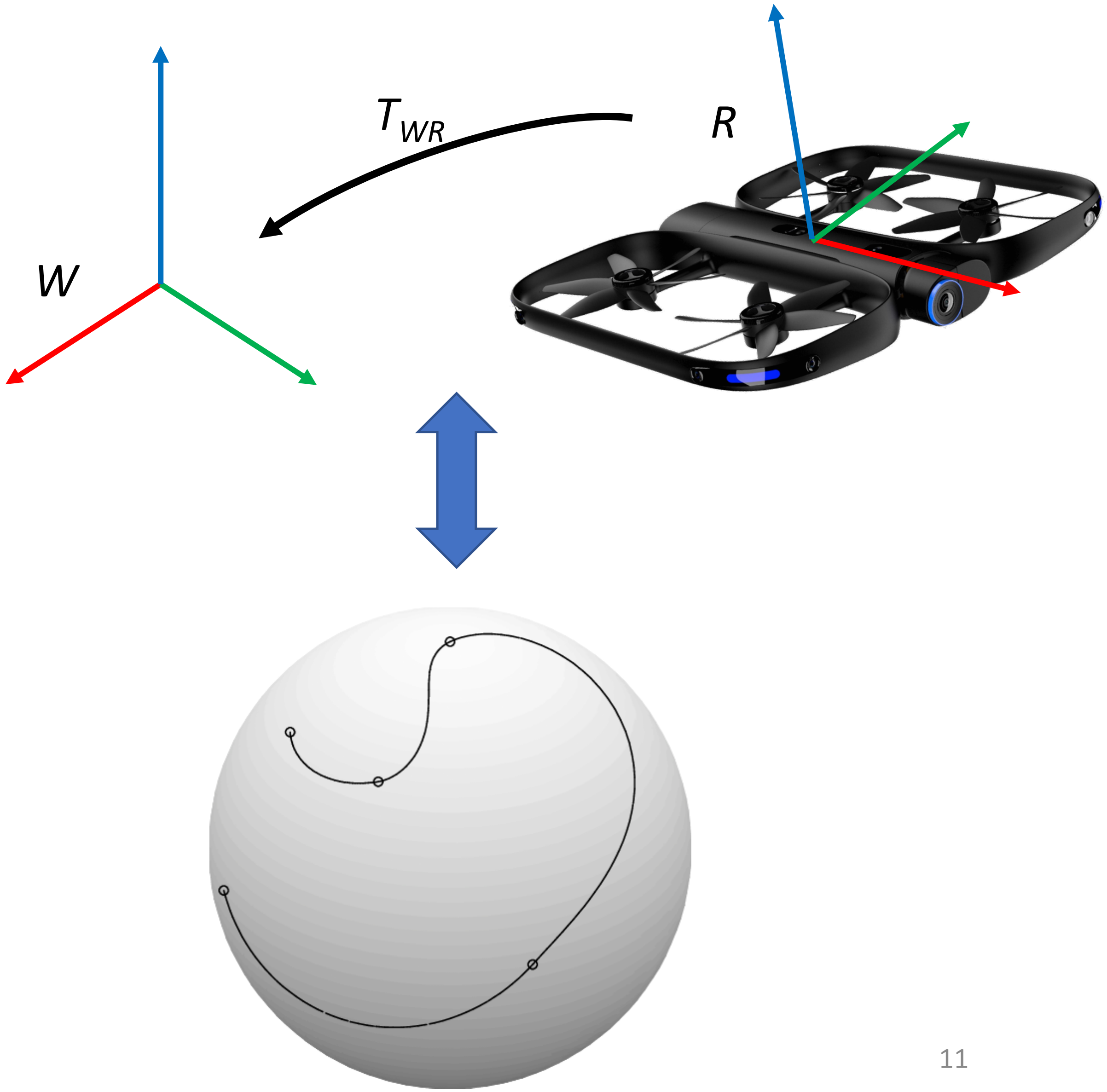
Motivating example: Matrix Lie groups (coordinate transformations)



Motivating application: Pose tracking

Recall that we model a robot's pose using the coordinate transformation from the robot's body-fixed frame *R* to the world frame *W*

$$T_{WR} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \in SE(3)$$



- SE(3) is a matrix Lie group:

$$SE(d) \coloneqq \left\{ \begin{pmatrix} R & b \\ 0 & 1 \end{pmatrix} \mid R \in SO(d), b \in \mathbb{R}^d \right\}$$

- Robot *pose* ↔ *point* in SE(3)
- Robot *velocity* ↔ *tangent vector* in SE(3)
- Robot *trajectory* ↔ *curve* in SE(3)

The left-multiplication map

To each $g \in G$ we associate the following smooth *left-multiplication map*:

$$L_g : G \to G$$
$$L_g(x) = gx.$$

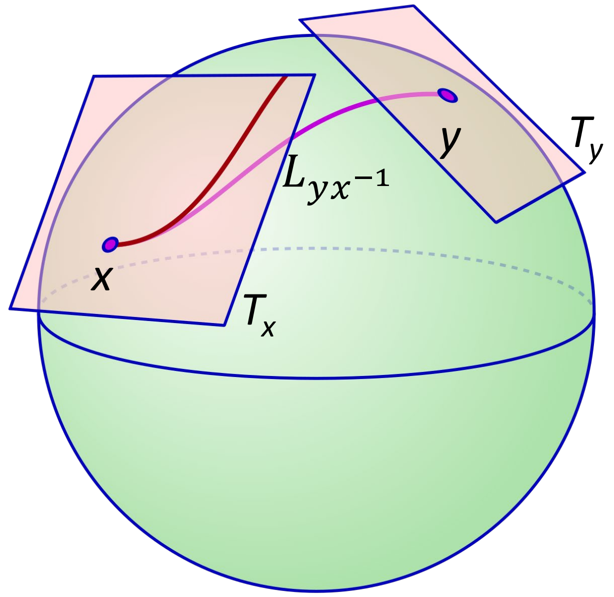
Key points:

- L_g is a smooth map, and it has a smooth inverse $L_{g^{-1}}$
  $\Rightarrow$ L_g is a *diffeomorphism* for all $g \in G$

- For any two points $x, y \in G$, $L_{yx^{-1}}(x) = y$
  $\Rightarrow$ There is a symmetry of G sending *any* point x onto *any* other point y

  $\Rightarrow$ G is a *homogeneous space*: all points $g \in G$ "look the same"

- Since L_x is a diffeomorphism, its derivative $dL_x : T_e(G) \to T_x(G)$ is an isomorphism for all x.
  $\Rightarrow$ We have a *canonical* way of identifying *every* tangent space $T_x(G)$ with the *single* "standard" tangent space $T_e(G)$ at the identity



The Lie Algebra and Left-Invariant Vector Fields

A *vector field* on a smooth manifold *M* is a smooth map *V* that assigns to each *x* in *M* a tangent vector $V(x) \in T_x(M)$.

A vector field *V* on a Lie group *G* is called *left-invariant* if:

$$dL_g V(x) = V(gx).$$

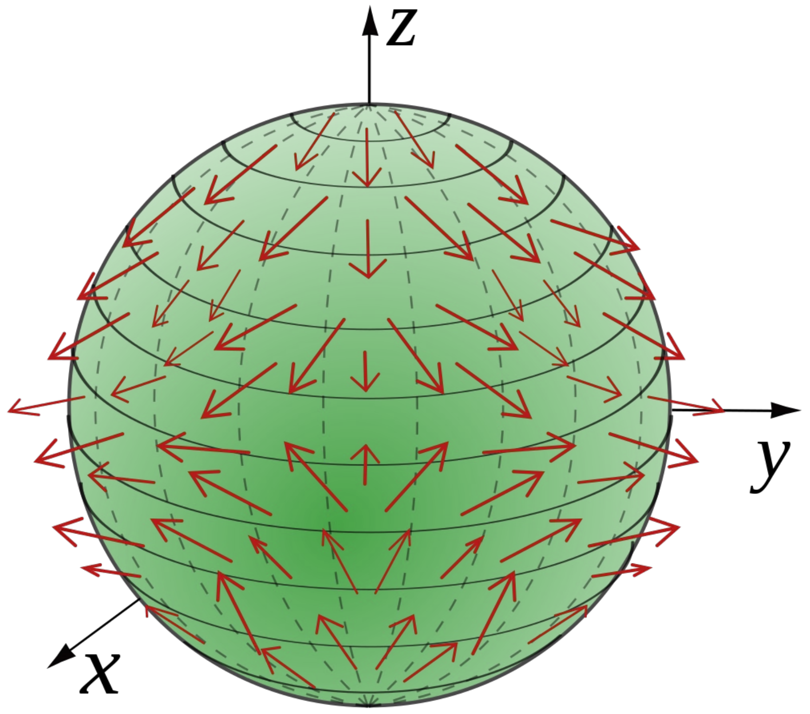
Key point: A left-invariant vector field is *completely determined* by its value at the identity element $e \in G$.

This gives an *identification*: $T_e(G) \Leftrightarrow$ {left-invariant vector fields on *G*}:

$$\omega \Leftrightarrow V_\omega$$

where V_ω is the left-invariant vector field defined by:

$$V_\omega(x) \triangleq dL_x(\omega)$$



Key point: We can identify a *velocity* $v \in T_x(M)$ at an *arbitrary* point $x \in G$ with a vector ω in the *fixed* tangent space $T_e(G)$.

We call the tangent space $T_e(G)$ the ***Lie algebra*** of *G*. We can think of this as the set of *infinitesimal generators for motion* on *G*.

The Exponential Map

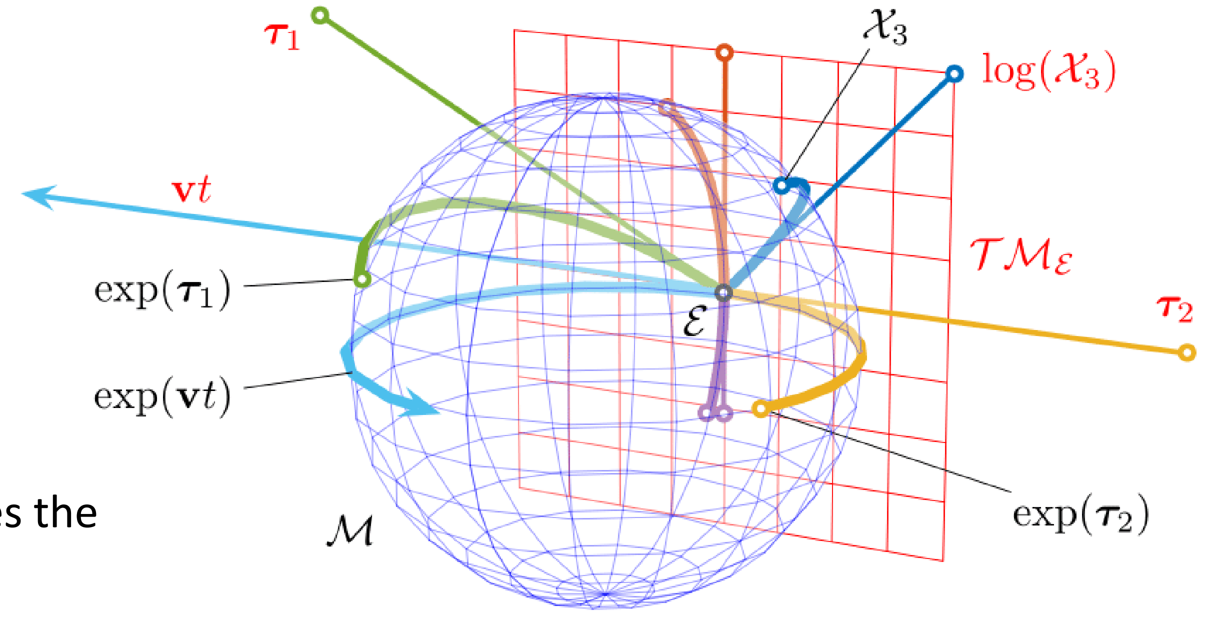
Recap: We just saw that every vector ω in $T_e(G)$ generates a left-invariant vector field on V_ω.

Key question: What is the *flow* on G generated by V_ω?

The *exponential map* provides the answer. This is a map

$$exp: T_e(G) \to G$$

from the Lie *algebra* into the Lie *group.*



Key property: The unique *integral curve* $c(t)$ of V_ω that takes the value x in G at time $t = 0$ is:

$$c(t) = x\exp(t\omega)$$

⇒ This tells us how to *move* on G along the flow of ω.

Special case: for *matrix groups* (subgroups of GL(n)), the exponential map has an especially nice form:

$$\exp(X) = \sum_{k=0}^{\infty} \frac{X^k}{k!}$$

⇒ This is how we "move around" on a matrix Lie group!

Lie Groups: Putting it all together

Main things to take away:

- Many objects of interest in robotics naturally belong to *non-Euclidean manifolds*, specifically *Lie groups*.
  Ex: Rotations, rigid motions, coordinate transforms, etc.

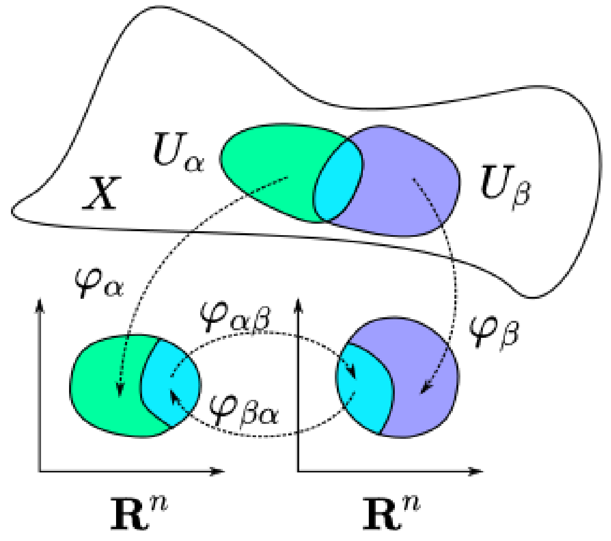
- Because manifolds are *locally* Euclidean, we can still (locally) stick coordinates on them and do calculus (yay ☺!)

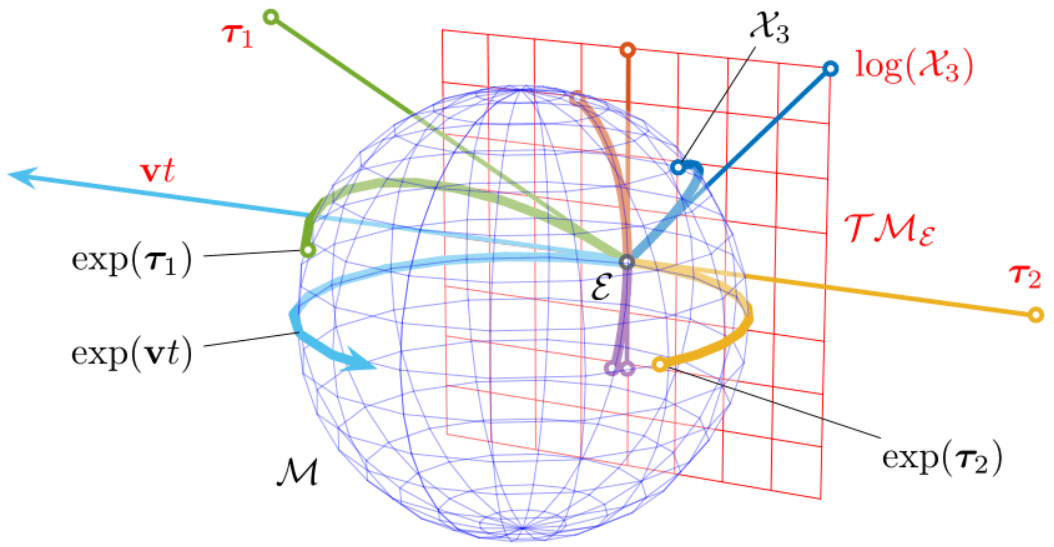
Lie groups are *especially* nice because:

- We can describe *velocities* *any* point using an element of the *Lie algebra* $T_e(G)$. This is *always* a *fixed* linear space (isomorphic to $\mathbb{R}^n$), no matter how complicated G is.

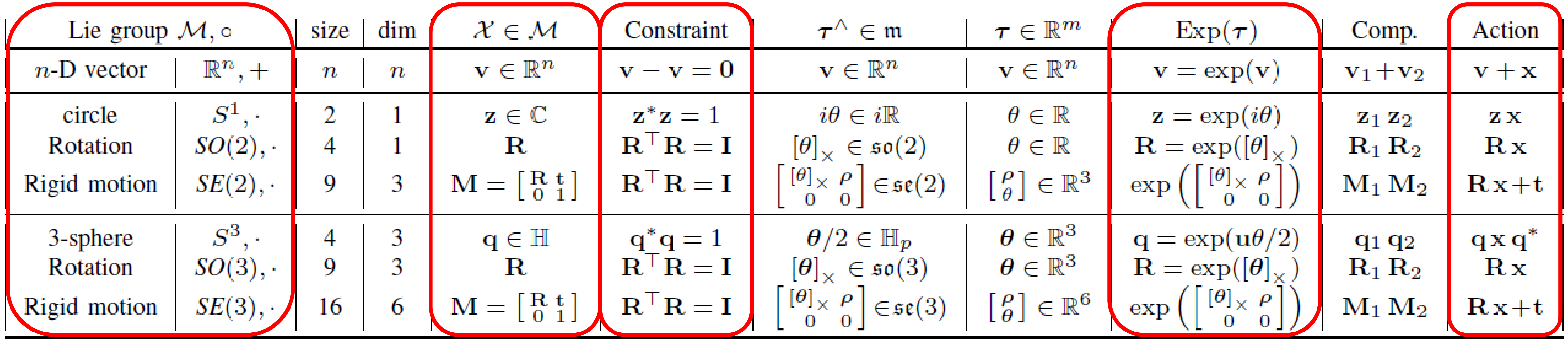
- We can *move* along a tangent vector ω using the group's *exponential map*. This mapping *automatically* respects the group's geometry (i.e. *satisfies constraints*).

Punchline: Lie group theory provides a *principled*, *unified language* for building algorithms on *arbitrary* Lie groups!





A Non-Exhaustive List of Commonly-Used Lie Groups in Robotics and Computer Vision



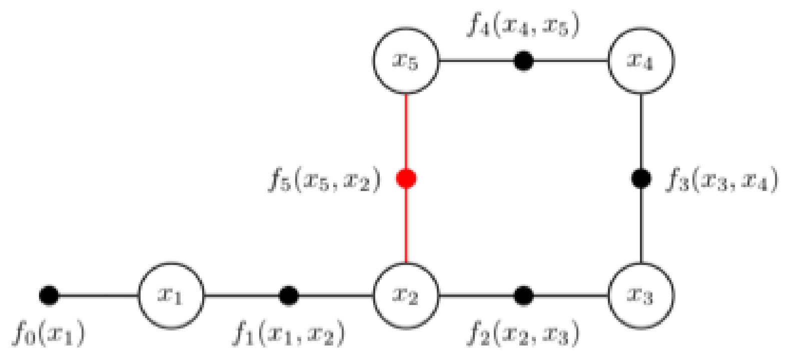
Lie group $\mathcal{M}, \circ$		size	dim	$\mathcal{X} \in \mathcal{M}$	Constraint	$\boldsymbol{\tau}^\wedge \in \mathfrak{m}$	$\boldsymbol{\tau} \in \mathbb{R}^m$	$\mathrm{Exp}(\boldsymbol{\tau})$	Comp.	Action
n-D vector	$\mathbb{R}^n, +$	n	n	$\mathbf{v} \in \mathbb{R}^n$	$\mathbf{v} - \mathbf{v} = \mathbf{0}$	$\mathbf{v} \in \mathbb{R}^n$	$\mathbf{v} \in \mathbb{R}^n$	$\mathbf{v} = \exp(\mathbf{v})$	$\mathbf{v}_1 + \mathbf{v}_2$	$\mathbf{v} + \mathbf{x}$
circle	$S^1, \cdot$	2	1	$\mathbf{z} \in \mathbb{C}$	$\mathbf{z}^* \mathbf{z} = 1$	$i\theta \in i\mathbb{R}$	$\theta \in \mathbb{R}$	$\mathbf{z} = \exp(i\theta)$	$\mathbf{z}_1 \, \mathbf{z}_2$	$\mathbf{z} \, \mathbf{x}$
Rotation	$SO(2), \cdot$	4	1	$\mathbf{R}$	$\mathbf{R}^\top \mathbf{R} = \mathbf{I}$	$[\theta]_\times \in \mathfrak{so}(2)$	$\theta \in \mathbb{R}$	$\mathbf{R} = \exp([\theta]_\times)$	$\mathbf{R}_1 \, \mathbf{R}_2$	$\mathbf{R} \, \mathbf{x}$
Rigid motion	$SE(2), \cdot$	9	3	$\mathbf{M} = \left[\begin{smallmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{smallmatrix}\right]$	$\mathbf{R}^\top \mathbf{R} = \mathbf{I}$	$\left[\begin{smallmatrix} [\theta]_\times & \boldsymbol{\rho} \\ 0 & 0 \end{smallmatrix}\right] \in \mathfrak{se}(2)$	$\left[\begin{smallmatrix} \boldsymbol{\rho} \\ \theta \end{smallmatrix}\right] \in \mathbb{R}^3$	$\exp\left(\left[\begin{smallmatrix} [\theta]_\times & \boldsymbol{\rho} \\ 0 & 0 \end{smallmatrix}\right]\right)$	$\mathbf{M}_1 \, \mathbf{M}_2$	$\mathbf{R} \, \mathbf{x} + \mathbf{t}$
3-sphere	$S^3, \cdot$	4	3	$\mathbf{q} \in \mathbb{H}$	$\mathbf{q}^* \mathbf{q} = 1$	$\boldsymbol{\theta}/2 \in \mathbb{H}_p$	$\boldsymbol{\theta} \in \mathbb{R}^3$	$\mathbf{q} = \exp(\mathbf{u}\theta/2)$	$\mathbf{q}_1 \, \mathbf{q}_2$	$\mathbf{q} \, \mathbf{x} \, \mathbf{q}^*$
Rotation	$SO(3), \cdot$	9	3	$\mathbf{R}$	$\mathbf{R}^\top \mathbf{R} = \mathbf{I}$	$[\boldsymbol{\theta}]_\times \in \mathfrak{so}(3)$	$\boldsymbol{\theta} \in \mathbb{R}^3$	$\mathbf{R} = \exp([\boldsymbol{\theta}]_\times)$	$\mathbf{R}_1 \, \mathbf{R}_2$	$\mathbf{R} \, \mathbf{x}$
Rigid motion	$SE(3), \cdot$	16	6	$\mathbf{M} = \left[\begin{smallmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{smallmatrix}\right]$	$\mathbf{R}^\top \mathbf{R} = \mathbf{I}$	$\left[\begin{smallmatrix} [\boldsymbol{\theta}]_\times & \boldsymbol{\rho} \\ 0 & 0 \end{smallmatrix}\right] \in \mathfrak{se}(3)$	$\left[\begin{smallmatrix} \boldsymbol{\rho} \\ \boldsymbol{\theta} \end{smallmatrix}\right] \in \mathbb{R}^6$	$\exp\left(\left[\begin{smallmatrix} [\boldsymbol{\theta}]_\times & \boldsymbol{\rho} \\ 0 & 0 \end{smallmatrix}\right]\right)$	$\mathbf{M}_1 \, \mathbf{M}_2$	$\mathbf{R} \, \mathbf{x} + \mathbf{t}$

J. Sola, J. Deray, D. Atchuthan: “A Micro Lie Theory for State Estimation in Robotics”

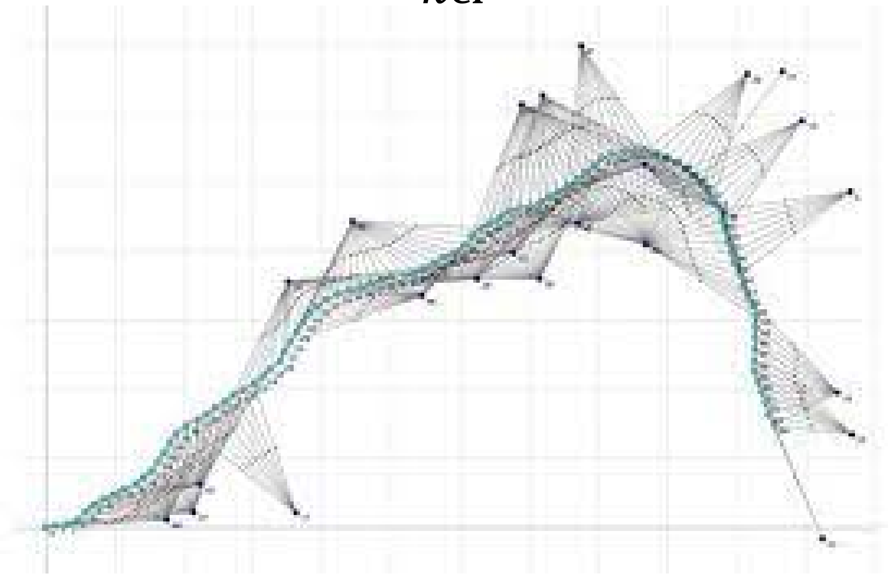
Application (preview): The GTSAM Library

- C++ / Python / MATLAB library for optimizing functions over Lie groups
- Functions are specified using *factor graphs*: graphical models describing how to build complex objectives from simple summands
- Makes it easy to model and solve complex robotic perception and control problems!
- **LOTS** of applications: Robotic mapping, visual-inertial odometry, 3D visual reconstruction, …

More on this later in the course …

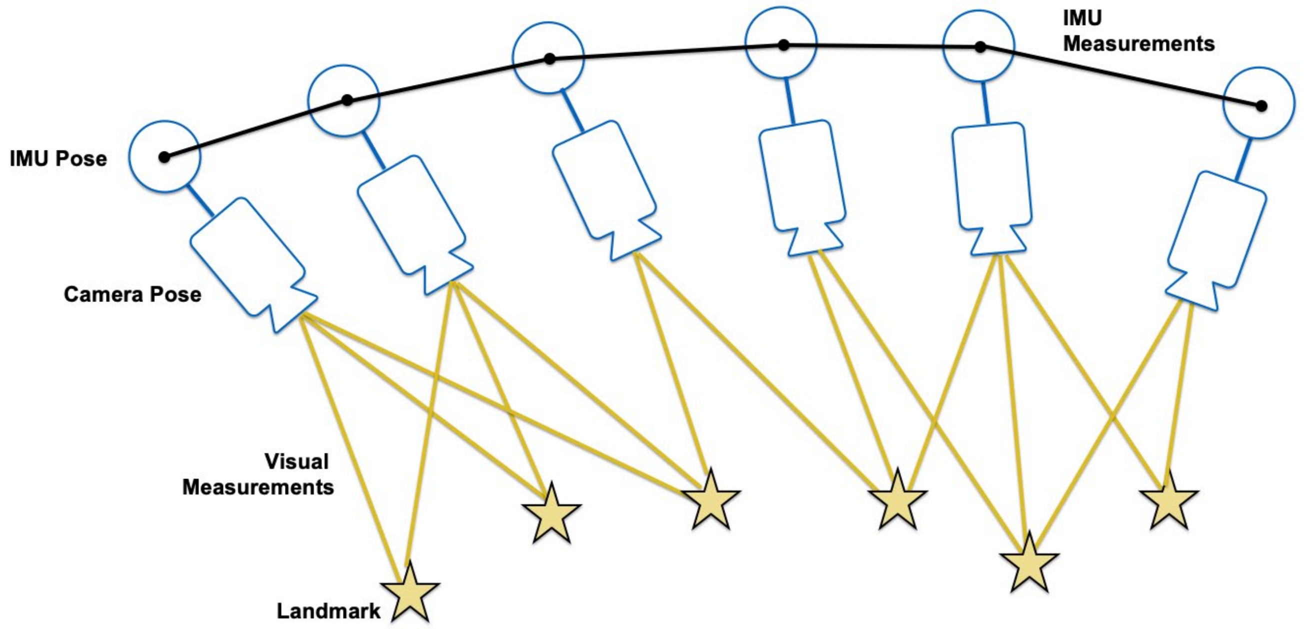


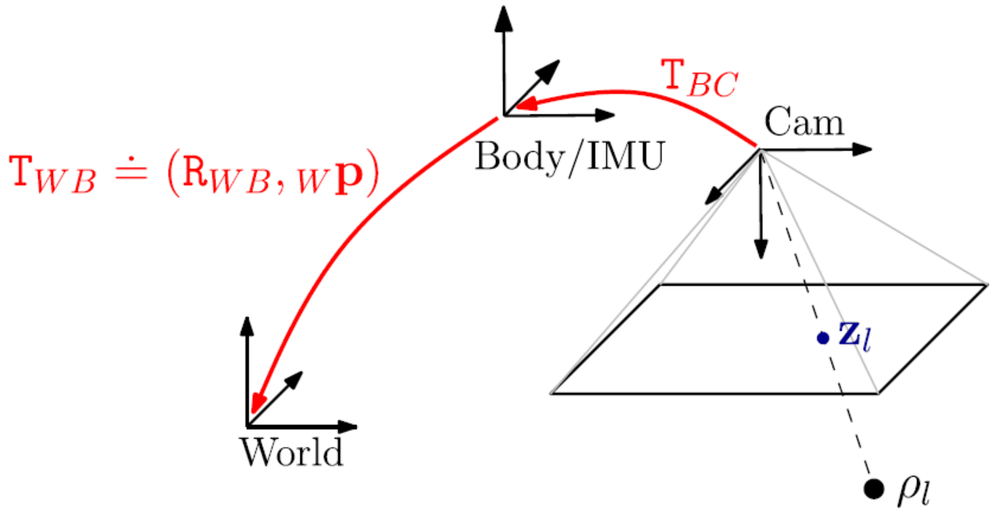
$$f(X) = \sum_{k \epsilon F} f_k(\Theta_k)$$



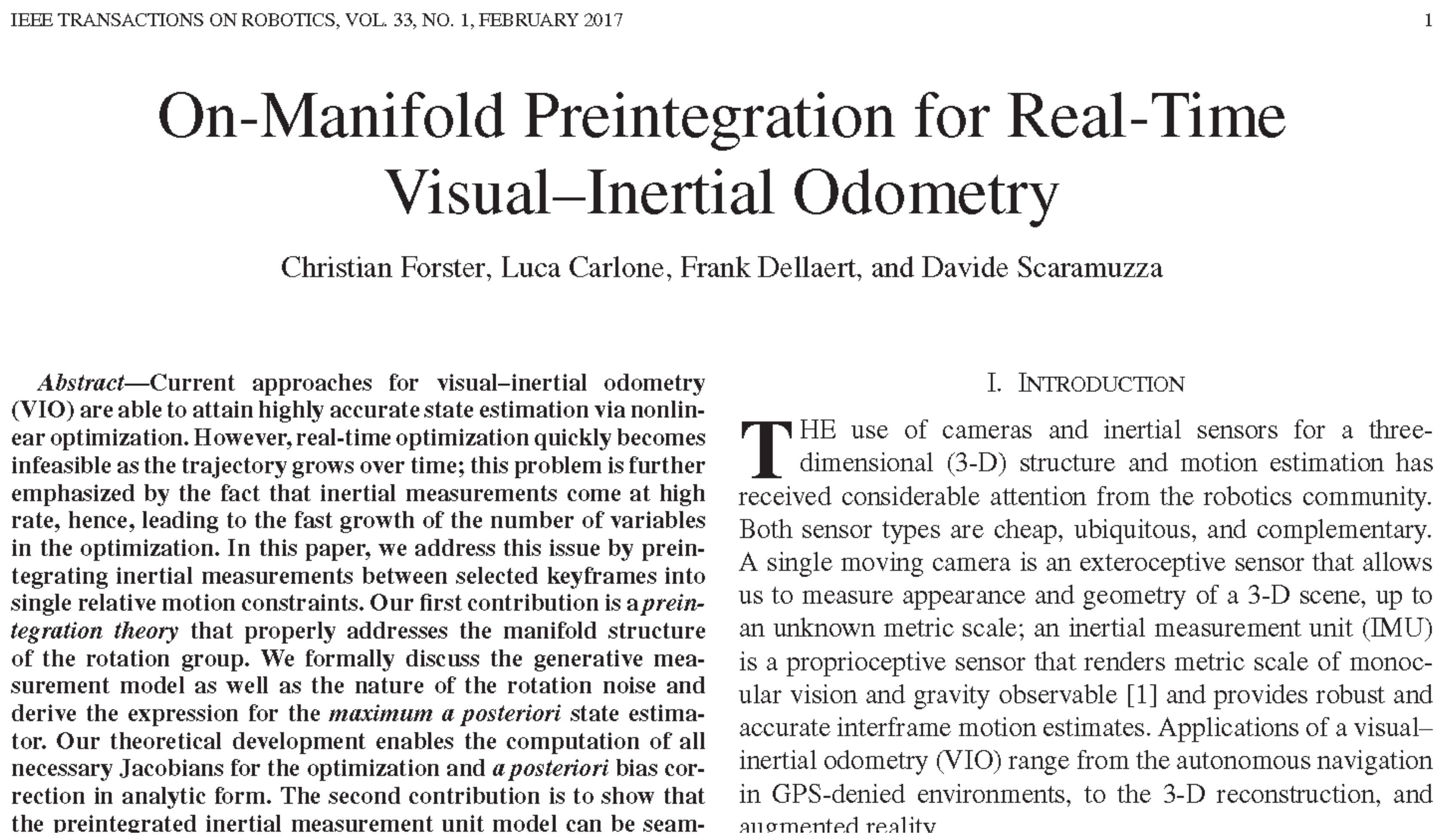
Example application: Visual-Inertial odometry

Main idea: Track the pose of a moving robot using a camera and inertial measurement unit (IMU)





Example: Visual-Inertial Odometry



IEEE TRANSACTIONS ON ROBOTICS, VOL. 33, NO. 1, FEBRUARY 2017 1

On-Manifold Preintegration for Real-Time Visual–Inertial Odometry

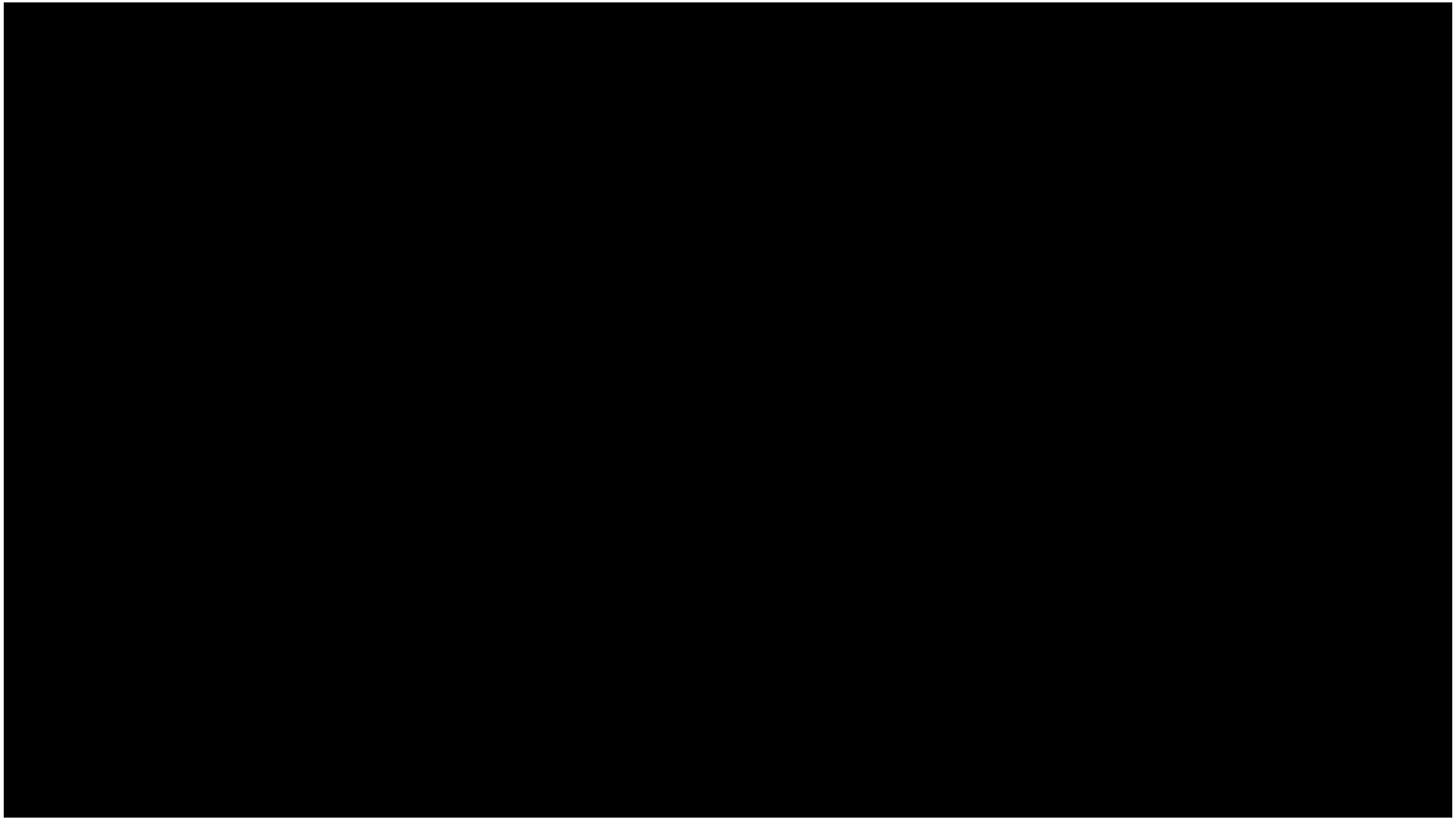
Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza

***Abstract*—Current approaches for visual–inertial odometry (VIO) are able to attain highly accurate state estimation via nonlinear optimization. However, real-time optimization quickly becomes infeasible as the trajectory grows over time; this problem is further emphasized by the fact that inertial measurements come at high rate, hence, leading to the fast growth of the number of variables in the optimization. In this paper, we address this issue by preintegrating inertial measurements between selected keyframes into single relative motion constraints. Our first contribution is a *preintegration theory* that properly addresses the manifold structure of the rotation group. We formally discuss the generative measurement model as well as the nature of the rotation noise and derive the expression for the *maximum a posteriori* state estimator. Our theoretical development enables the computation of all necessary Jacobians for the optimization and *a posteriori* bias correction in analytic form. The second contribution is to show that the preintegrated inertial measurement unit model can be seam-**

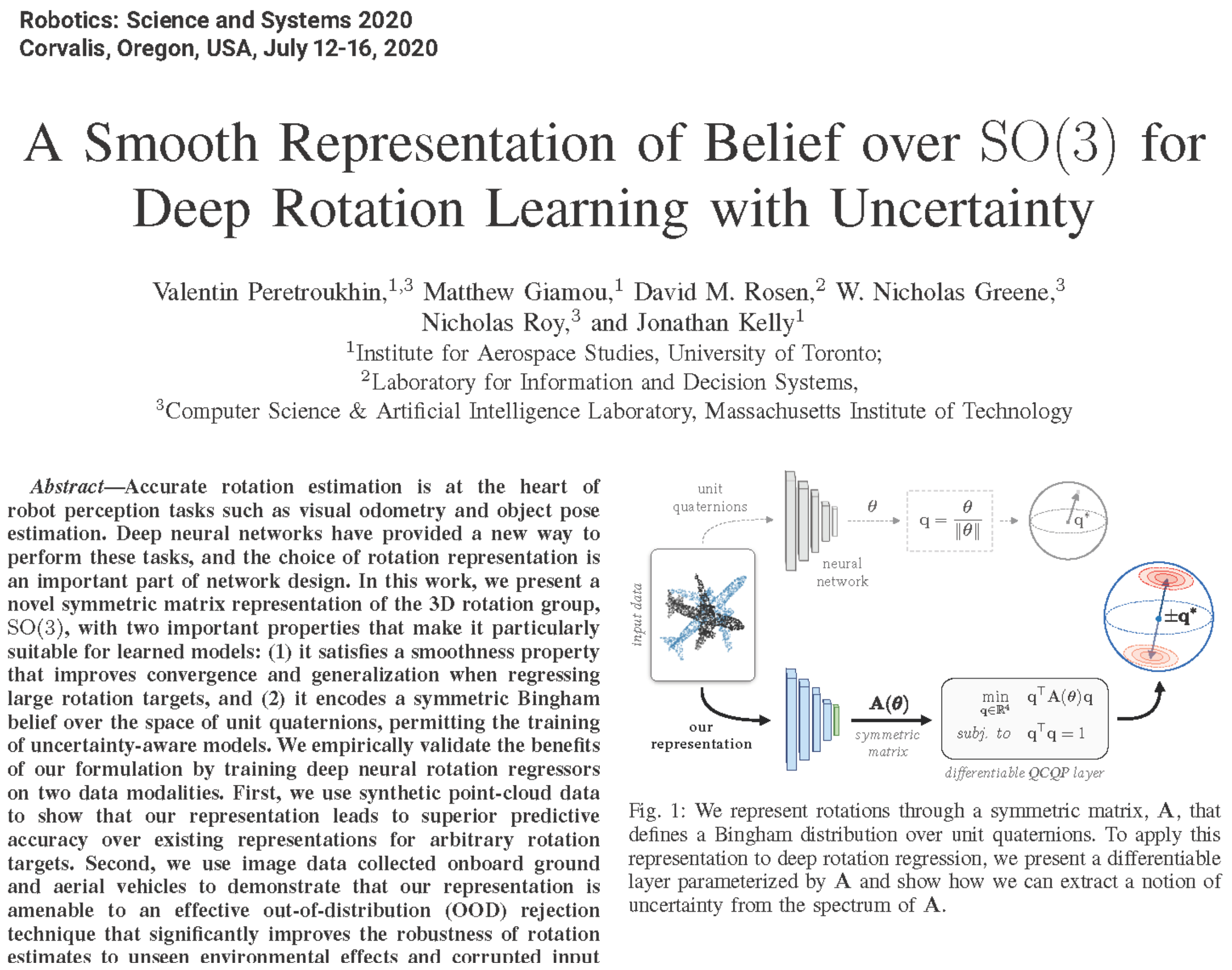
I. INTRODUCTION

THE use of cameras and inertial sensors for a three-dimensional (3-D) structure and motion estimation has received considerable attention from the robotics community. Both sensor types are cheap, ubiquitous, and complementary. A single moving camera is an exteroceptive sensor that allows us to measure appearance and geometry of a 3-D scene, up to an unknown metric scale; an inertial measurement unit (IMU) is a proprioceptive sensor that renders metric scale of monocular vision and gravity observable [1] and provides robust and accurate interframe motion estimates. Applications of a visual–inertial odometry (VIO) range from the autonomous navigation in GPS-denied environments, to the 3-D reconstruction, and augmented reality

Example: Visual-Inertial Odometry



Example: Learning to Estimate Rotations



Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

A Smooth Representation of Belief over $\mathrm{SO}(3)$ for Deep Rotation Learning with Uncertainty

Valentin Peretroukhin,[1,3] Matthew Giamou,[1] David M. Rosen,[2] W. Nicholas Greene,[3]
Nicholas Roy,[3] and Jonathan Kelly[1]
[1]Institute for Aerospace Studies, University of Toronto;
[2]Laboratory for Information and Decision Systems,
[3]Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology

***Abstract*—Accurate rotation estimation is at the heart of robot perception tasks such as visual odometry and object pose estimation. Deep neural networks have provided a new way to perform these tasks, and the choice of rotation representation is an important part of network design. In this work, we present a novel symmetric matrix representation of the 3D rotation group, $\mathrm{SO}(3)$, with two important properties that make it particularly suitable for learned models: (1) it satisfies a smoothness property that improves convergence and generalization when regressing large rotation targets, and (2) it encodes a symmetric Bingham belief over the space of unit quaternions, permitting the training of uncertainty-aware models. We empirically validate the benefits of our formulation by training deep neural rotation regressors on two data modalities. First, we use synthetic point-cloud data to show that our representation leads to superior predictive accuracy over existing representations for arbitrary rotation targets. Second, we use image data collected onboard ground and aerial vehicles to demonstrate that our representation is amenable to an effective out-of-distribution (OOD) rejection technique that significantly improves the robustness of rotation estimates to unseen environmental effects and corrupted input**

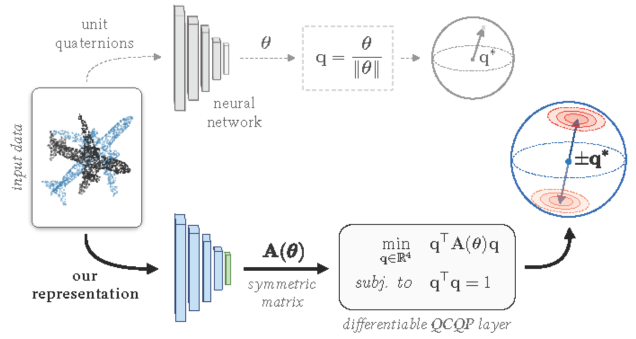
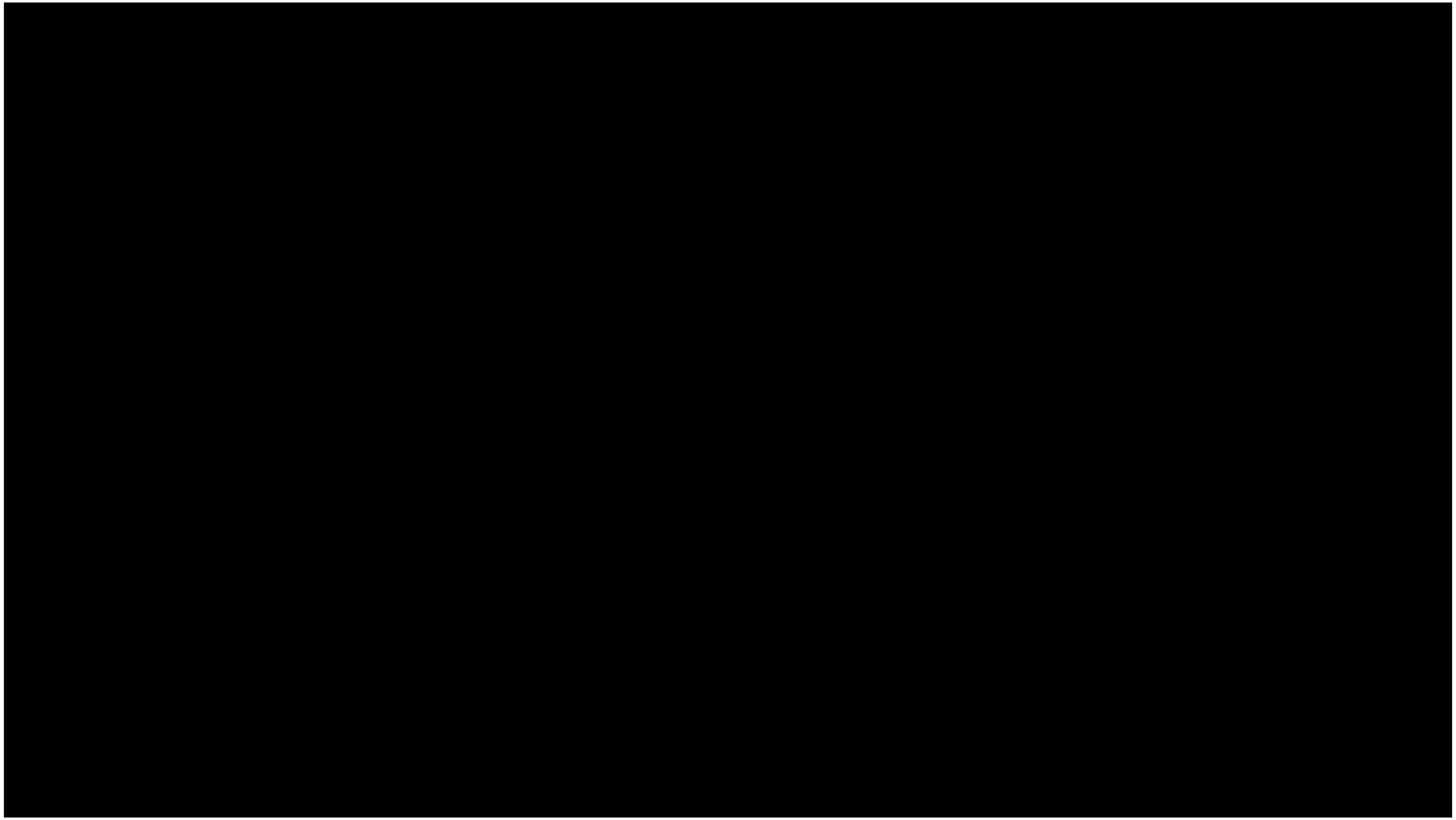


Fig. 1: We represent rotations through a symmetric matrix, $\mathbf{A}$, that defines a Bingham distribution over unit quaternions. To apply this representation to deep rotation regression, we present a differentiable layer parameterized by $\mathbf{A}$ and show how we can extract a notion of uncertainty from the spectrum of $\mathbf{A}$.

Example: Learning to Estimate Rotations



Lie Groups: Putting it all together

Main things to take away:

- Many objects of interest in robotics naturally belong to *non-Euclidean manifolds*, specifically *Lie groups*.
  Ex: Rotations, rigid motions, coordinate transforms, etc.

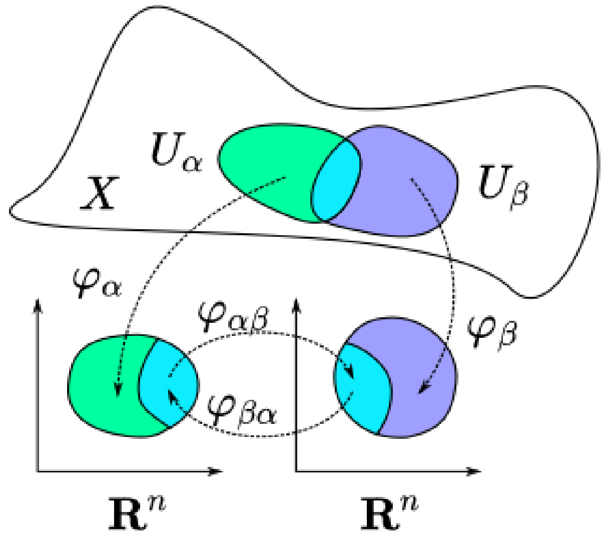
- Because manifolds are *locally* Euclidean, we can still (locally) stick coordinates on them and do calculus (yay ☺!)

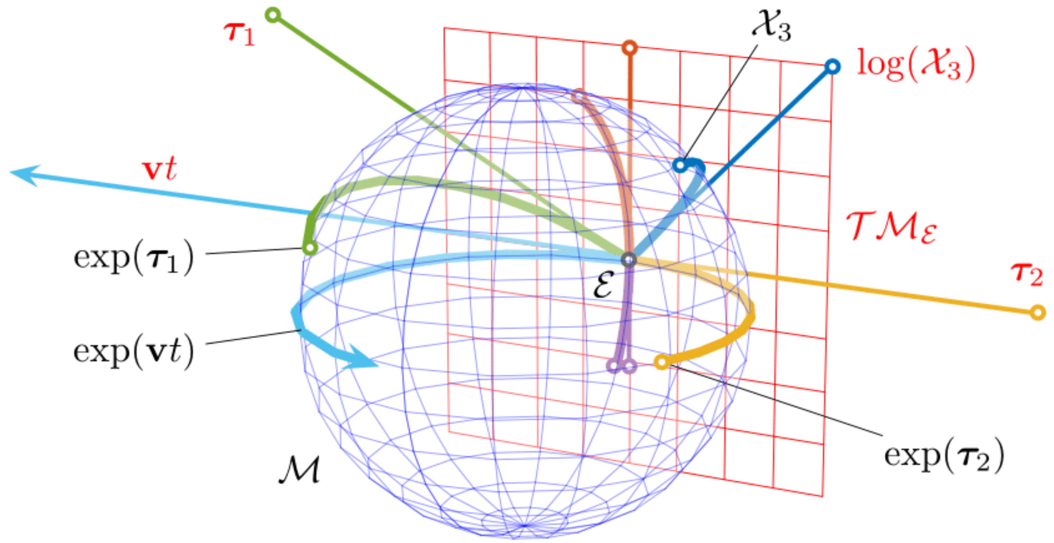
Lie groups are *especially* nice because:

- We can describe *velocities* *any* point using an element of the *Lie algebra* $T_e(G)$. This is *always* a *fixed* linear space (isomorphic to $\mathbb{R}^n$), no matter how complicated *G* is.

- We can *move* along a tangent vector ω using the group's *exponential map*. This mapping *automatically* respects the group's geometry (i.e. *satisfies constraints*).

Punchline: Lie group theory provides a *principled*, *unified language* for building algorithms on *arbitrary* Lie groups!


X
U_α
U_β
φ_α
$\varphi_{\alpha\beta}$
φ_β
$\varphi_{\beta\alpha}$
$\mathbf{R}^n$
$\mathbf{R}^n$


τ_1
$\mathcal{X}_3$
$\log(\mathcal{X}_3)$
$\mathbf{v}t$
$\mathcal{TM}_\mathcal{E}$
$\exp(\tau_1)$
$\mathcal{E}$
τ_2
$\exp(\mathbf{v}t)$
$\mathcal{M}$
$\exp(\tau_2)$