Non-Euclidean manifolds in robotics and computer vision: why should we care?

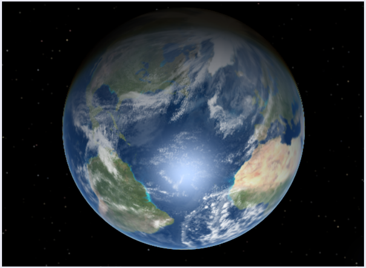
Jose Luis Blanco Claraco

Universidad de Almería
http://www.ual.es/~jlblanco/

March 18th, 2013
Universidad de Zaragoza

Short answer for the impatient

Non-Euclidean manifolds in robotics and computer vision: why should we care? $\rightarrow$ Because we can't ignore the *real geometry* of the mathematical spaces!



"What is the distance from Madrid to Miami?"

Contents

Contents

Concept: topological spaces

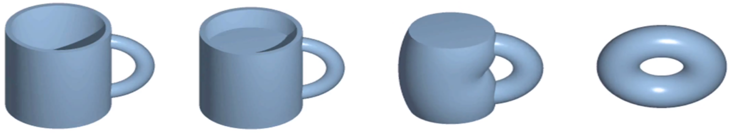
The topology of an object defines its “shape” in the coarsest sense: no geometric (“metric”) details.

Concept: topological spaces

The topology of an object defines its “shape” in the coarsest sense: no geometric (“metric”) details.

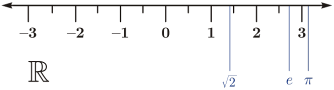
Two objects are topologically the same if we can morph one into another just streching, without creating new holes (*tearing*) or closing existing ones.

Classic example: a coffee mug and a torus.

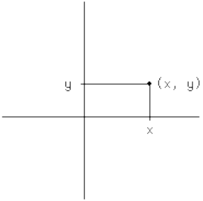


Topological spaces: examples

- The line of real numbers $\mathbb{R}$.



- The infinite plane of $\mathbb{R}^2$.



Topological spaces: examples

- The circle S^1.
- The torus is the Cartesian product of two circles: $S^1 \times S^1$
- The sphere is the different space S^2.

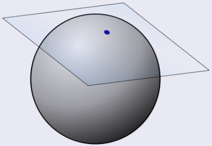
Concept: What is a manifold?

An N-dimensional **manifold** M is a topological space where every point $p \in M$ is endowed with *local* Euclidean structure.

Informally: a manifold is built by "gluing" together small pieces of $\mathbb{R}^N$. However, the "global" structure is not Euclidean.

Example

A sphere is a 2D manifold (*surface*) embedded in $\mathbb{R}^3$:

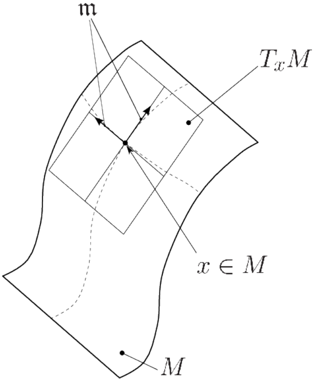


Concept: manifold tangent space

Concept: manifold tangent space

The **tangent space** of the N-dimensional manifold M at $x \in M$ can be seen as the local Euclidean space at x. Also called the "linearization" of the manifold.

Denoted as $T_x M$, is the vector space of all possible "velocities" of x:



Concept: Lie group

A **Lie group** is a (non-empty) subset G of $\mathbb{R}^N$ that fulfills:

1. G is a group.
2. G is a manifold in $\mathbb{R}^N$.
3. Both, the group product operation $(\cdot : G \mapsto G)$ and its inverse $(\cdot^{-1} : G \mapsto G)$ are smooth functions.

Concept: Lie group

A **Lie group** is a (non-empty) subset G of $\mathbb{R}^N$ that fulfills:

1. G is a group.
2. G is a manifold in $\mathbb{R}^N$.
3. Both, the group product operation $(\cdot : G \mapsto G)$ and its inverse $(\cdot^{-1} : G \mapsto G)$ are smooth functions.

Matrix Lie groups

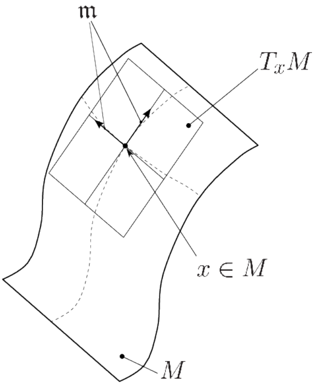
1. Lie groups of our interest are *matrix spaces*.
2. A theorem due to Von Newman and Cartan [Bla10]:

   A closed subgroup G of $\mathbf{GL}(N, \mathbb{R})$ is a linear Lie group $\rightarrow$ is also a smooth manifold in $\mathbb{R}^{N^2}$

Concept: Lie algebra

Concept: Lie algebra

For any Lie group, its **Lie algebra** $\mathfrak{m}$ is the set of base vectors of its tangent space at the identity $\mathbf{I}$: $\mathfrak{m} = T_{\mathbf{I}}M$



An example: the Lie group SE(3) (I)

SE(3) is the group of all 4×4 matrices for rigid translations + rotations:

An example: the Lie group SE(3) (I)

SE(3) is the group of all 4×4 matrices for rigid translations + rotations:

$$SE(3) = \left\{ \mathbf{M} \;\middle|\; \mathbf{M} = \left[\begin{array}{ccc|c} & & & x \\ & \mathbf{R}_{3\times3} & & y \\ & & & z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \right\}, \text{with } \mathbf{R}_{3\times3} \in SO(3)$$

An example: the Lie group SE(3) (I)

SE(3) is the group of all 4×4 matrices for rigid translations + rotations:

$$SE(3) = \left\{ \mathbf{M} \;\middle|\; \mathbf{M} = \left[\begin{array}{ccc|c} & & & x \\ & \mathbf{R}_{3\times3} & & y \\ & & & z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \right\}, \text{with } \mathbf{R}_{3\times3} \in SO(3)$$

Clarifying the number of dimensions

- Environment space dimensions $= 4 \times 4 = 16$
- Manifold dimensions $= 6$ (DOFs)

An example: the Lie group SE(3) (I)

SE(3) is the group of all 4×4 matrices for rigid translations + rotations:

$$SE(3) = \left\{ \mathbf{M} \;\middle|\; \mathbf{M} = \left[\begin{array}{ccc|c} & & & x \\ & \mathbf{R}_{3\times 3} & & y \\ & & & z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \right\}, \text{with } \mathbf{R}_{3\times 3} \in SO(3)$$

Its Lie algebra $\mathfrak{se}(3)$ is a *vector* base, whose elements are 6 *matrices*:

$$\mathbf{G}_1^{\mathfrak{se}(3)} = \left(\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \right) \qquad \mathbf{G}_4^{\mathfrak{se}(3)} = \left(\begin{array}{ccc|c} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \right)$$

$$\mathbf{G}_2^{\mathfrak{se}(3)} = \left(\begin{array}{ccc|c} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \right) \qquad \mathbf{G}_5^{\mathfrak{se}(3)} = \left(\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \right)$$

$$\mathbf{G}_3^{\mathfrak{se}(3)} = \left(\begin{array}{ccc|c} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \right) \qquad \mathbf{G}_6^{\mathfrak{se}(3)} = \left(\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array} \right)$$

An example: the Lie group SE(3) (II)

OK, but... WHY? Where do those skew symmetric matrices come from?!

An example: the Lie group SE(3) (II)

OK, but... WHY? Where do those skew symmetric matrices come from?!

Remember: Lie algebra is a *vector base* ("matrix base", actually) of the tangent space, which is the space of all possible "**velocities**" on the manifold.

An example: the Lie group SE(3) (II)

OK, but... WHY? Where do those skew symmetric matrices come from?!

Remember: Lie algebra is a *vector base* ("matrix base", actually) of the tangent space, which is the space of all possible "**velocities**" on the manifold.

Take derivatives of the transformation matrix with respect to x, y, z, yaw ϕ, pitch χ and roll ψ at the identity and you're there!

$$\mathbf{R}(\phi,\chi,\psi) = \mathbf{R}_z(\phi)\mathbf{R}_y(\chi)\mathbf{R}_x(\psi)$$
$$= \begin{pmatrix} \cos\phi\cos\chi & \cos\phi\sin\chi\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\chi\cos\psi + \sin\phi\sin\psi \\ \sin\phi\cos\chi & \sin\phi\sin\chi\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\chi\cos\psi - \cos\phi\sin\psi \\ -\sin\chi & \cos\chi\sin\psi & \cos\chi\cos\psi \end{pmatrix}$$

$$\left.\frac{\partial \mathbf{R}}{\partial \phi}\right|_{\phi=0,\chi=0,\psi=0} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}_\times$$

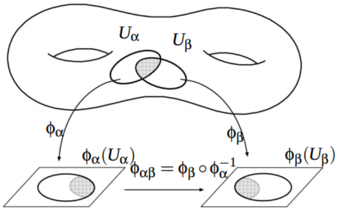
etc.

Concept: Charts

Concept: Charts

A **chart** is the bijective parameterization of (part of) a N-dimensional manifold into $\mathbb{R}^N$. They enable doing calculus (e.g. derivatives) on manifolds.

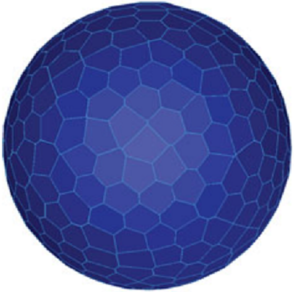
Example: two overlapping charts of a double torus:



(Image credits: [GHQ06])

Concept: Atlas

An **atlas** is a collection of charts that covers the entire manifold – nomenclature as in "real-world" Cartography!.



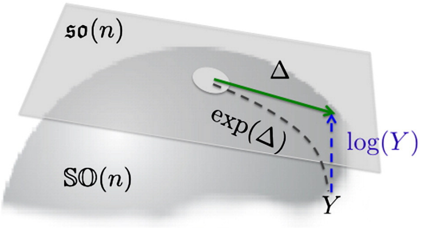
(Image credits: [GHQ06])

Concept: Exponential & logarithm maps

Maps for a Lie group M

- **Exponential map**: A map from the local tangent space $T_x M \rightarrow$ the manifold M.
- **Logarithm map**: Inverse map, from the manifold $M \rightarrow T_x M$.

Note: The projections of Lie algebra's vector base through the exponential are tangent to the **geodesics** on the manifold at x.



(Image credits: [Lui12])

Contents

Normalized histograms

Topology

If $\mathbf{x} \in \mathbb{R}_{+}^{M}$ is a normalized histogram, the set of all of them:

$$\mathbb{P}^{M-1} = \left\{ x_1, \ldots, x_M \,\middle|\, \sum_{m=1}^{M} x_m = 1, x_m \geq 0 \right\} \tag{1}$$

is the *simplex* $\mathbb{P}^{M-1}$, a manifold with corners.

Normalized histograms

How are *distances* defined over normalized histograms

- Geodesics are good-old straight lines! $\rightarrow$ L1, L2 norms and alike are OK.
- But... from machine learning there is much more to say: learning significative χ^2 distances, etc. [AS11].

Normalized histograms

How are *distances* defined over normalized histograms

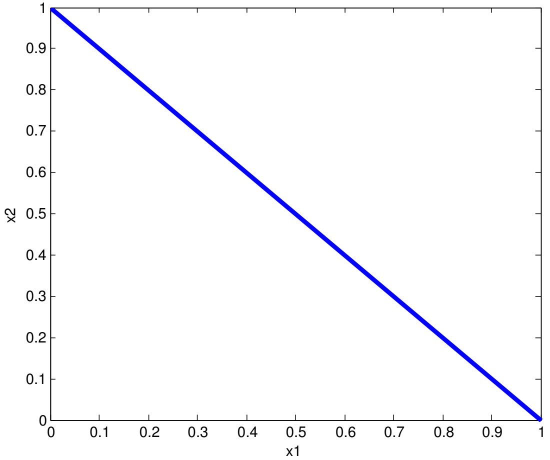
- Geodesics are good-old straight lines! $\rightarrow$ L1, L2 norms and alike are OK.
- But... from machine learning there is much more to say: learning significative χ^2 distances, etc. [AS11].

Optimizing / derivatives over histograms?

Here, the topology does matter. Let's see it graphically...

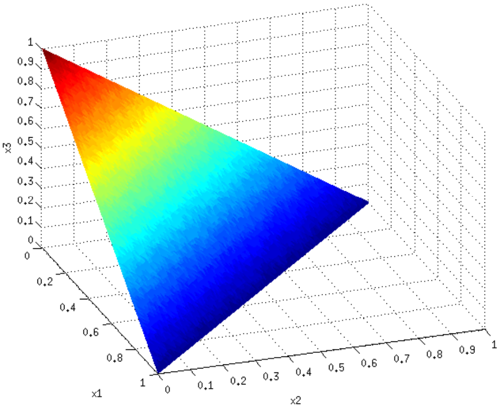
Normalized histograms: M=2

For M=2, (x_1, x_2), we have a 1-simplex (a segment):



Normalized histograms: M=3

For M=3, (x_1, x_2, x_3), we have a 2-simplex (a triangle):



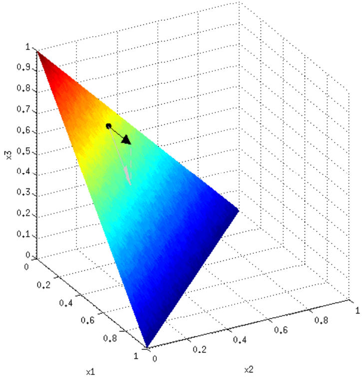
Normalized histograms: M=3

The Jacobian for some target function $\mathbf{f}$:

$$\frac{d\mathbf{f}}{d\{x_1, x_2, x_3\}} = \begin{bmatrix} \frac{d\mathbf{f}}{dx_1} & \frac{d\mathbf{f}}{dx_2} & \frac{d\mathbf{f}}{dx_3} \end{bmatrix}$$

may give us a gradient direction $\notin$ the manifold.

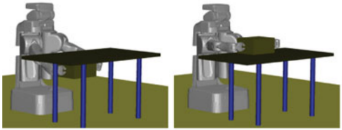
We'll see how to properly deal with optimizations on manifolds in a few minutes...



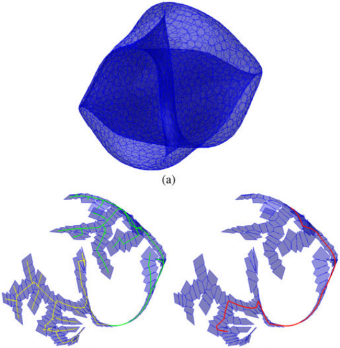
Contents

Sampling for Rapidly-Exploring Random Trees (RRTs)

A *graph* can be built to represent the topology of the atlas (remember: atlas=collection of charts) of kinematically-constrained problems:



(3) PR2 with a box



(a) (b) (c)

(Image credits: [GHQ06])

Contents

Principal Component Analysis (PCA)

PCA

The standard PCA method:

- Given a set of N-dimensional samples, determine the q directions of "principal variation".
- It can be solved by finding the *eigenvectors* of the covariance of data points, and keeping those with the q largest eigenvalues.

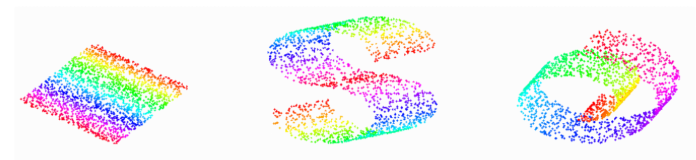
Principal Component Analysis (PCA)

PCA

The standard PCA method:

- Given a set of N-dimensional samples, determine the q directions of "principal variation".
- It can be solved by finding the *eigenvectors* of the covariance of data points, and keeping those with the q largest eigenvalues.

It works fine for Euclidean spaces, but couldn't handle non-linear manifolds:



Works OK — Does not — Does not

(Image credits: [Ihl03])

PCA on manifolds

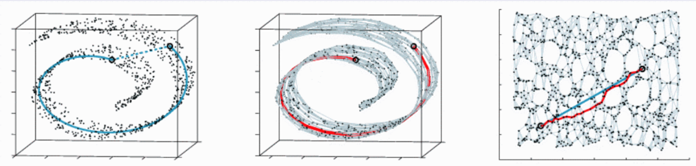
The key difference

- In PCA we want to **maximize** the variance of the largest components → an implicit **metric** for distances (Euclidean norm).

The key difference

- In PCA we want to **maximize** the variance of the largest components $\rightarrow$ an implicit **metric** for distances (Euclidean norm).
- On manifolds, distances $\rightarrow$ distances along **geodesics**.

Use methods like **IsoMap** [TdSL00], to "unroll" manifolds:



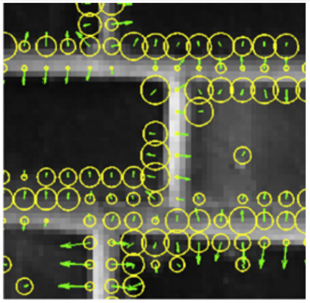
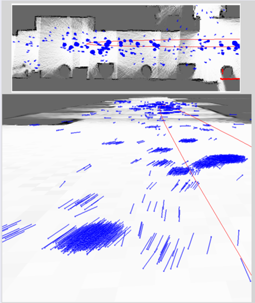
(Image credits: [Ihl03])

Contents

Motivation

We may need to average, for example:

- Orientations of image gradients, blob-like features, etc.
- Estimating the most-likely heading from a set of particles in Monte-Carlo localization.

(Image credits: [XHJF12])

Averaging in SO(2)

The Special Orthogonal Group in 2D: SO(2)

- From all 2×2 invertible matrices GL(2), only a few represent *rigid*, pure rotations in the 2D plane.

Averaging in SO(2)

The Special Orthogonal Group in 2D: SO(2)

- From all 2×2 invertible matrices GL(2), only a few represent *rigid*, pure rotations in the 2D plane.
- Orthonormal matrices $\mathbf{R} = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \rightarrow$ we need *two* numbers to unequivocally define a rotation, since $(a, b) \cdot (c, d) = 0$ (and $|\mathbf{R}| > 0) \Rightarrow (c, d) = (-b, a)$, so:

$$\mathbf{R}(a, b) = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$$

Averaging in SO(2)

The Special Orthogonal Group in 2D: SO(2)

This group of matrices is isomorphic to S^1: a circle, with 1 DOF:

$$\mathbf{R}(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

Averaging in SO(2)

The Special Orthogonal Group in 2D: SO(2)

This group of matrices is isomorphic to S^1: a circle, with 1 DOF:

$$\mathbf{R}(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

$$\rightarrow SO(2) = \begin{cases} \text{A manifold with } \mathbf{1} \text{ DOF,} \\ \text{but needs } \mathbf{2} \text{ different numbers for representing!} \end{cases}$$

Think it this way: there is no way to map S^1 to a segment of $\mathbb{R}^1$ and preserve the circular topology.

Example 1

Put it clear:

$$\underbrace{\text{Orientations in 2D}}_{\text{A manifold with } S^1 \text{ topology}} \neq \underbrace{\text{Values for the parameter } \phi}_{\text{This parameter lives in } \mathbf{R}^1}$$

Example 1

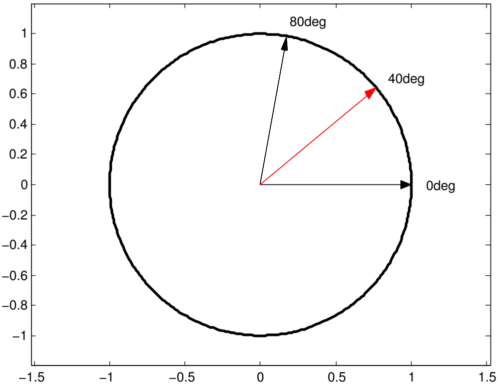
Put it clear:

$$\underbrace{\text{Orientations in 2D}}_{\text{A manifold with } S^1 \text{ topology}} \neq \underbrace{\text{Values for the parameter } \phi}_{\text{This parameter lives in } \mathbf{R}^1}$$

Let's see the practical implications...

Example 1

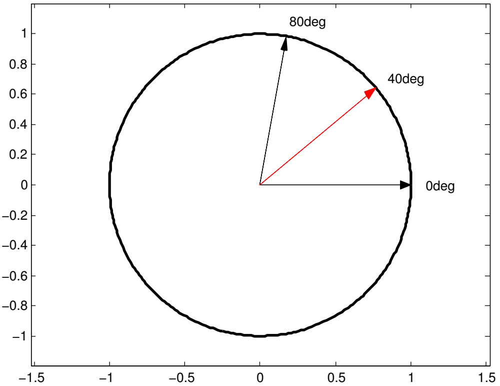
What is the mean orientation of 0° and 80°?



Arithmetic mean of 0° and 80° $= 40^\circ$ (the correct mean) $\rightarrow$ Intuitive!

Example 1

What is the mean orientation of 0° and 80°?



Arithmetic mean of 0° and $80^\circ = 40^\circ$ (the correct mean) $\rightarrow$ Intuitive!
...but is only correct "by accident" for averages of **only two values**.

Mean

But... what is **actually** the mean or average of a set of values?

Mean

But... what is **actually** the mean or average of a set of values?

Common definition:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$$

Mean

But... what is **actually** the mean or average of a set of values?

Common definition:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$$

It turns out that this is just a *special case* for Euclidean spaces! Everything depends on the **metric** for defining distances.

Meaning of "mean"

A generic (*invariant*) definition of "mean"

Given a **metric** $d(\mathbf{x}, \mathbf{y})$, the average of $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ is:

$$\bar{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N} d(\mathbf{p} - \mathbf{p}_i)^2$$

Meaning of “mean”

A generic (*invariant*) definition of “mean”

Given a **metric** $d(\mathbf{x}, \mathbf{y})$, the average of $\{\mathbf{p}_1, ..., \mathbf{p}_N\}$ is:

$$\bar{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N} d(\mathbf{p} - \mathbf{p}_i)^2$$

On manifolds, distances are measured over **geodesics**, the “straight lines” of curved spaces.

Note: In Euclidean $\mathbb{R}^M$, geodesics are good-old straight lines!

Metrics for matrix spaces

The point is: distances are not measured for values of the parameter ϕ, but *on the manifold* $\mathrm{SO}(2)$.

Metrics for matrix spaces

The point is: distances are not measured for values of the parameter ϕ, but *on the manifold* SO(2).

A standard metric for matrix spaces $\rightarrow$ Frobenius norm:

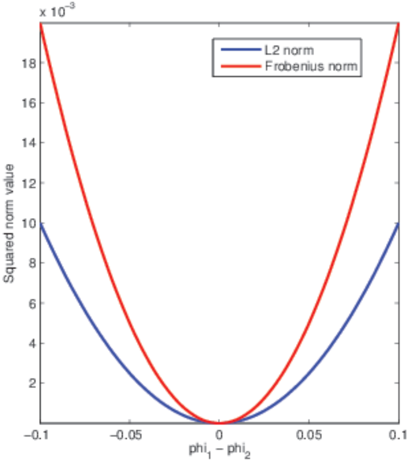
$$||\mathbf{A}||_F^2 = \sum_i \sum_j a_{ij}^2 = tr(\mathbf{A}\mathbf{A}^\top)$$

Metrics on SO(2)

It's clear both metrics are different for the distance between orientations ϕ_1 and ϕ_2:

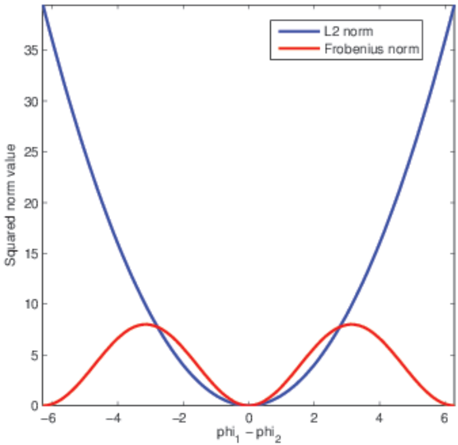
Operating, one gets:

$$d(\phi_1, \phi_2)_F^2 = 4\left[1 - \cos(\phi_1 - \phi_2)\right]$$



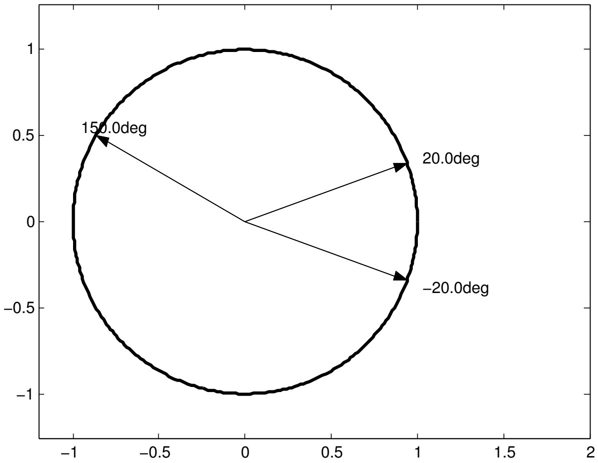
Metrics on SO(2)

It's clear both metrics are different for the distance between orientations ϕ_1 and ϕ_2:



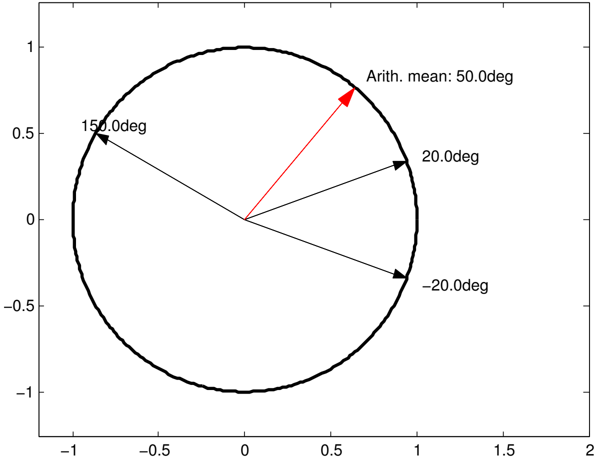
Example 2

Let's show this with a new example:



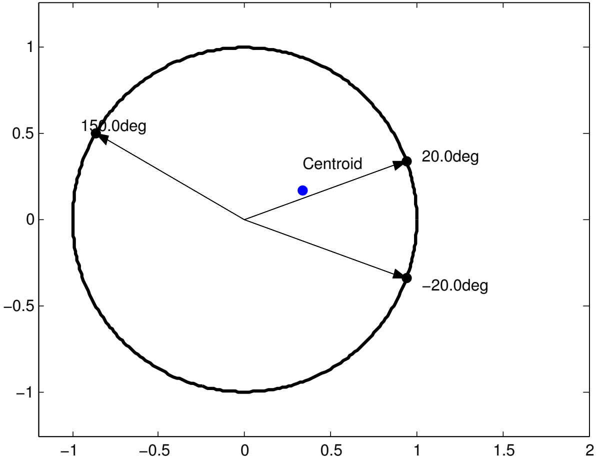
Example 2

The (WRONG) average from arithmetic mean of ϕ is 50°...



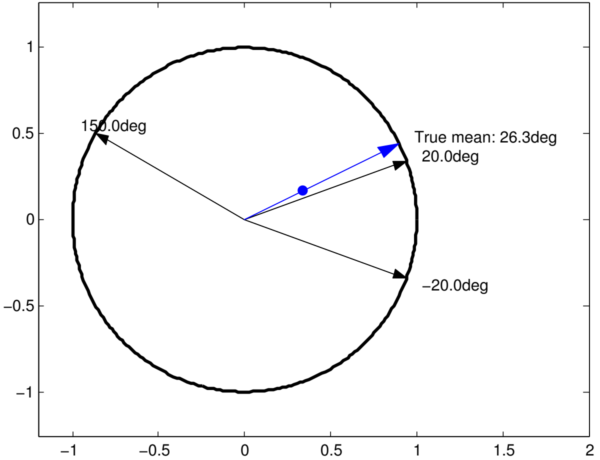
Example 2

Instead: (1) evaluate the centroid of all 2×2 matrices,



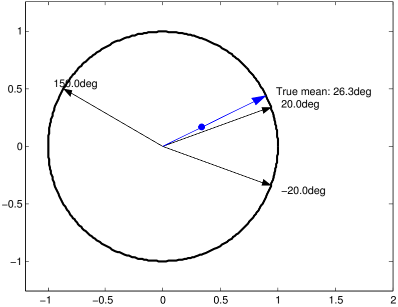
Example 2

and (2) **project** the point onto the manifold. $26.3° \neq 50°$!!



Example 2

and (2) **project** the point onto the manifold. $26.3^\circ \neq 50^\circ$!!



Why the **centroid** of 2D points? $\rightarrow$ think of the first column in SO(2) matrices...

Contents

Two important manifolds

SO(3)

The Lie group of 3D rotations:

$$SO(3) = \left\{ \mathbf{R} \in GL(3, \mathbb{R}) \,\middle|\, \mathbf{R}^\top \mathbf{R} = \mathbf{I}, |\mathbf{R}| = 1 \right\}$$

SE(3)

The Lie group of 3D rigid transformations (4×4 matrices):

$$SE(3) = \underbrace{SO(3)}_{\text{Rotation}} \times \underbrace{\mathbb{R}^3}_{\text{Translation}}$$

The meaning of mean (once more)

Means and metrics

The *variational* definition of *mean* involves a definition of *distances* on the manifold:

$$\bar{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N} d(\mathbf{p} - \mathbf{p}_i)^2$$

Advantage wrt the $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ definition: it is **invariant**.

The meaning of mean (once more)

Means and metrics

The *variational* definition of *mean* involves a definition of *distances* on the manifold:

$$\bar{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N} d(\mathbf{p} - \mathbf{p}_i)^2$$

Advantage wrt the $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ definition: it is **invariant**.

The notion of "mean" is not obvious for manifolds and there exist **as many different "means" as metrics**.

Two means for SO(3)

(1) Euclidean mean

- Using the Frobenius norm:
  $d_F(\mathbf{R} - \mathbf{R}_i) = ||\mathbf{R} - \mathbf{R}_i||_F = tr(\mathbf{R}^\top \mathbf{R}_i)$

Two means for SO(3)

(1) Euclidean mean

- Using the Frobenius norm: $d_F(\mathbf{R} - \mathbf{R}_i) = ||\mathbf{R} - \mathbf{R}_i||_F = tr(\mathbf{R}^\top \mathbf{R}_i)$
- Can be shown to be equivalent to the previous 2D example: (1) "centroid" of SO(3) matrices, then (2) project to SO(3) – e.g. doable via Singular Value Decomposition (SVD).

Two means for SO(3)

(1) Euclidean mean

- Using the Frobenius norm: $d_F(\mathbf{R} - \mathbf{R}_i) = ||\mathbf{R} - \mathbf{R}_i||_F = tr(\mathbf{R}^\top \mathbf{R}_i)$
- Can be shown to be equivalent to the previous 2D example: (1) "centroid" of SO(3) matrices, then (2) project to SO(3) – e.g. doable via Singular Value Decomposition (SVD).

(2) Riemannian mean

- Using the Riemannian distance $d_R(\mathbf{R} - \mathbf{R}_i) = \frac{1}{\sqrt{2}} || \log(\mathbf{R}^\top \mathbf{R}_i) ||_F$
- It stands for the length of the shortest geodesic between two matrices.

(See [Moa02] for more details and closed-form formulas)

What about SE(3)?

- Unlike for SO(3), there exists no bi-invariant metric in SE(3)
- Still, good metrics for computing means exist: [SWR10]

Contents

Independent vs. dependent coordinates

Problem

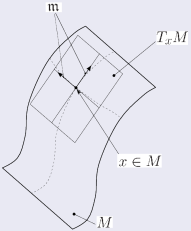
Integrate or minimize:

$$f(\mathbf{q})$$

for $\mathbf{q} \in \mathbb{R}^n$, restricted to $\mathbf{q} \in M$, with M an m-dimensional manifold ($m < n$), defined as $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$.

Coordinates

- Dependent coordinates (dims$=n$): $\mathbf{q}$.
- Independent coordinates (dims$=m < n$): $\mathbf{z} \in T_x M$.



(1) The Lagrange multiplier method

Augmented problem with $n - m$ new unknowns $\boldsymbol{\lambda}$ (Lagrange multipliers):

$$\left.\begin{array}{c} f(\mathbf{q}) \\ \boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0} \end{array}\right\} \rightarrow f(\mathbf{q}) + \frac{\partial \Phi(\mathbf{q})}{\partial \mathbf{q}}^{\top} \boldsymbol{\lambda}$$

(1) The Lagrange multiplier method

Augmented problem with $n - m$ new unknowns $\boldsymbol{\lambda}$ (Lagrange multipliers):

$$\left.\begin{array}{c} f(\mathbf{q}) \\ \boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0} \end{array}\right\} \rightarrow f(\mathbf{q}) + \underbrace{\frac{\partial \boldsymbol{\Phi}(\mathbf{q})}{\partial \mathbf{q}}^{\top} \boldsymbol{\lambda}}_{\text{This should be zero}}$$

Idea: Lagrange multipliers can always be found that make the second term vanish.

(1) The Lagrange multiplier method

Augmented problem with $n-m$ new unknowns $\boldsymbol{\lambda}$ (Lagrange multipliers):

$$\left.\begin{array}{c} f(\mathbf{q}) \\ \boldsymbol{\Phi}(\mathbf{q}) = \mathbf{0} \end{array}\right\} \to f(\mathbf{q}) + \underbrace{\frac{\partial \Phi(\mathbf{q})}{\partial \mathbf{q}}^{\top} \boldsymbol{\lambda}}_{\text{This should be zero}}$$

Idea: Lagrange multipliers can always be found that make the second term vanish.
This method is widely-used in **dynamical simulations** of kinematically-constrained robots, mechanisms, etc.

(2) The tangent space method

Basic idea

Replace "global" optimizations with local solutions on the tangent space.

(2) The tangent space method

Least-squares optimization

$$\delta_k^\star \leftarrow \left. \frac{\partial F(x_{k-1} + \delta_k)}{\partial \delta_k} \right|_{\delta_k} = 0 \qquad \Rightarrow \qquad x_k \leftarrow x_{k-1} + \delta_k^\star$$

where $\delta \in$ ambient space, "$+$" is the standard Euclidean addition.

(2) The tangent space method

Least-squares optimization

$$\delta_k^\star \leftarrow \left. \frac{\partial F(x_{k-1} + \delta_k)}{\partial \delta_k} \right|_{\delta_k} = 0 \qquad \Rightarrow \qquad x_k \leftarrow x_{k-1} + \delta_k^\star$$

where $\delta \in$ ambient space, "$+$" is the standard Euclidean addition.

On-manifold least-squares

$$\epsilon_k^\star \leftarrow \left. \frac{\partial F(x_{k-1} \boxplus \epsilon_k)}{\partial \epsilon_k} \right|_{\epsilon=0} = 0 \qquad \Rightarrow \qquad x_k \leftarrow x_{k-1} \boxplus \epsilon_k^\star$$

where $\epsilon \in$ manifold, $\boxplus$ the Lie group operation (i.e. matrix multiplication)

(2) The tangent space method

Example:

Given an observation model $h(\mathbf{p})$ for $\mathbf{p} \in \mathbb{R}^3$ the relative location of a landmark, in SLAM we find:

$$h(\mathbf{p}) = h(\mathbf{L} \ominus \mathbf{x}) \quad \rightarrow \quad \begin{cases} \mathbf{L} \in \mathbb{R}^3 : \text{landmark coordinates,} \\ \mathbf{x} \in SE(3) : \text{camera pose.} \end{cases}$$

(2) The tangent space method

Example:

Given an observation model $h(\mathbf{p})$ for $\mathbf{p} \in \mathbb{R}^3$ the relative location of a landmark, in SLAM we find:

$$h(\mathbf{p}) = h(\mathbf{L} \ominus \mathbf{x}) \quad \rightarrow \quad \begin{cases} \mathbf{L} \in \mathbb{R}^3 : \text{landmark coordinates,} \\ \mathbf{x} \in SE(3) : \text{camera pose.} \end{cases}$$

During optimization, we find the Jacobian:

$$\frac{\partial h(\mathbf{L} \ominus (\mathbf{x} + \boldsymbol{\Delta}_x))}{\partial \boldsymbol{\Delta}_x} \quad \text{(Euclidean)}$$

(2) The tangent space method

Example:

Given an observation model $h(\mathbf{p})$ for $\mathbf{p} \in \mathbb{R}^3$ the relative location of a landmark, in SLAM we find:

$$h(\mathbf{p}) = h(\mathbf{L} \ominus \mathbf{x}) \quad \rightarrow \quad \begin{cases} \mathbf{L} \in \mathbb{R}^3 : \text{landmark coordinates,} \\ \mathbf{x} \in SE(3) : \text{camera pose.} \end{cases}$$

During optimization, we find the Jacobian:

$$\frac{\partial h(\mathbf{L} \ominus (\mathbf{x} + \boldsymbol{\Delta}_x))}{\partial \boldsymbol{\Delta}_x} \quad \text{(Euclidean)}$$

$$\frac{\partial h(\mathbf{L} \ominus (\mathbf{x} \oplus \boldsymbol{\epsilon}))}{\partial \boldsymbol{\epsilon}} \quad \text{(On-manifold SE(3))}$$

(2) The tangent space method

Example:

Given an observation model $h(\mathbf{p})$ for $\mathbf{p} \in \mathbb{R}^3$ the relative location of a landmark, in SLAM we find:

$$h(\mathbf{p}) = h(\mathbf{L} \ominus \mathbf{x}) \quad \rightarrow \quad \begin{cases} \mathbf{L} \in \mathbb{R}^3 : \text{landmark coordinates,} \\ \mathbf{x} \in SE(3) : \text{camera pose.} \end{cases}$$

During optimization, we find the Jacobian:

$$\frac{\partial h(\mathbf{L} \ominus (\mathbf{x} + \mathbf{\Delta}_x))}{\partial \mathbf{\Delta}_x} \quad \text{(Euclidean)}$$

$$\frac{\partial h(\mathbf{L} \ominus (\mathbf{x} \oplus \epsilon))}{\partial \epsilon} \quad \text{(On-manifold SE(3))}$$

Trick: Apply the chain rule representing poses and points as homogeneous 4×4 matrices $\rightarrow$ compositions become matrix multiplications $\rightarrow$ simple Jacobians!!

(2) The tangent space method

Historical remarks

- ∼1820?: Could be traced to Gauss' works on survey
- ...
- 1994: Taylor & Kriegman proposal for SO(3) [TK94]

(2) The tangent space method

Historical remarks

- $\sim$1820?: Could be traced to Gauss' works on survey
- ...
- 1994: Taylor & Kriegman proposal for SO(3) [TK94]
- (Re?–)Introduction in the SLAM community (AFAIK):
  - 2006: Mentioned in a German work by Udo Frese et al. [FSH^{+}].
  - 2008: First work in English is [Her08], a Bachelor's thesis by Christoph Hertzberg, adviced by Udo Frese.

Applications to EKF

Applicability

Has been introduced in graph-SLAM, but is a **general framework**!

Applications to EKF

Motivation

- EKF with quaternions has been quite common and successful in visual SLAM.

Applications to EKF

Motivation

- EKF with quaternions has been quite common and successful in visual SLAM.
- but the filter does not respect the **normalization** of the quaternion $\rightarrow$ need to renormalize after each step.
- In theory, it's sub-optimal (though, I haven't tested this numerically!)

Applications to EKF

Motivation

- EKF with quaternions has been quite common and successful in visual SLAM.
- but the filter does not respect the **normalization** of the quaternion $\rightarrow$ need to renormalize after each step.
- In theory, it's sub-optimal (though, I haven't tested this numerically!)

EKF with on-manifold derivatives

Required changes:

- Choose a parameterization (quaternion is OK here!)
- Replace all Jacobians ($\frac{\partial \cdot}{\partial \Delta_x} \rightarrow \frac{\partial \cdot}{\partial \epsilon}$)
- Use manifold Jacobian to update the EKF mean,
- Apply chain rule to get the correct Jacobian that updates the covariance in parameterization space, not the manifold.

See detailed formulas, etc. [Bla10, FMB12]

Contents

References I

Vitaly Ablavsky and Stan Sclaroff, *Learning parameterized histogram kernels on the simplex manifold for image and action classification*, Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1473–1480.

Jose-Luis Blanco, *A tutorial on se (3) transformation parameterizations and on-manifold optimization*, Tech. report, Technical report, University of Malaga, 2010.

Juan-Antonio Fernández-Madrigal and José-Luis Blanco, *Simultaneous localization and mapping for mobile robots: Introduction and methods*, IGI Global, sep 2012.

Udo Frese, Lutz Schröder, Christoph Hertzberg, Janosch Machowinski, and René Wagner, *Theorie der sensorfusion*.

Xianfeng Gu, Ying He, and Hong Qin, *Manifold splines*, Graphical Models **68** (2006), no. 3, 237–254.

Christoph Hertzberg, *A framework for sparse, non-linear least squares problems on manifolds*, 2008.

A. Ihler, *Nonlinear Manifolds Learning (Part one)*, Tech. report, MIT, 2003.

References II

- Yui Man Lui, *Advances in matrix manifolds for computer vision*, Image and Vision Computing **30** (2012), no. 6, 380–388.
- Maher Moakher, *Means and averaging in the group of rotations*, SIAM journal on matrix analysis and applications **24** (2002), no. 1, 1–16.
- Inna Sharf, Alon Wolf, and M.B. Rubin, *Arithmetic and geometric solutions for average rigid-body rotation*, Mechanism and Machine Theory (2010).
- Tenenbaum, de Silva, and Langford, *Isomap*, 2000.
- Camillo J Taylor and David J Kriegman, *Minimization on the lie group so (3) and related manifolds*, Yale University (1994).
- Yong Xu, Sibin Huang, Hui Ji, and Cornelia Fermüller, *Scale-space texture description on sift-like textons*, Computer Vision and Image Understanding (2012).

Non-Euclidean manifolds in robotics and computer vision: why should we care?

Jose Luis Blanco Claraco

Universidad de Almería
http://www.ual.es/~jlblanco/

March 18th, 2013
Universidad de Zaragoza

Thank you for your attention!