Syntax	Input	Output	Identities
Lie group operations			
$c = a.\mathbf{compose}(b)$ $c = a{*}b$	a in g b in a	$c = b$ in g $c = ab$	
$b = a.\mathbf{inverse}()$	a in g	$b = g$ in a $b = a^{-1}$	$a.\mathbf{compose}($ $a.\mathbf{inverse}()) == \mathbf{I}$
$c = a.\mathbf{between}(b)$	a in g b in g	$c = b$ in a $c = a^{-1}b$	$a.\mathbf{inverse}().$ $\mathbf{compose}(b) == c$
$\delta = a.\mathbf{logmap}()$	a in g	δ in g $\hat{\delta} = \log a$	$\mathrm{X} :: \mathbf{Expmap}(\delta) == a$
$a = \mathrm{X} :: \mathbf{Expmap}(\delta)$	δ in g	a in g $a = \exp \hat{\delta}$	$a.\mathbf{logmap}() == \delta$
Lie group actions			
$q = a.\mathbf{transform\_to}(p)$	a in g p in g	$q = p$ in a $q = a^{-1}p$	
$q = a.\mathbf{transform\_from}(p)$ $q = a{*}p$	a in g p in a	$q = p$ in g $q = ap$	

Table 1: Coordinate frame transformations performed by GTSAM geometry operations. Here, a, b, c, and g are Lie group elements (Pose2, Pose3, Rot2, Rot3, Point2, Point3, *etc*). δ is a set of Lie algebra coordinates (i.e. linear update, linear delta, tangent space coordinates), and X is a Lie group type (e.g. Pose2). p and q are the objects of Lie group actions (Point2, Point3, *etc*).

1 Introduction

This document describes the coordinate frame conventions in which GTSAM inputs and represents states and uncertainties. When specifying initial conditions, measurements and their uncertainties, and interpreting estimated uncertainties and the results of geometry operations, the coordinate frame convention comes into play.

GTSAM as consistently as possible represents all states and uncertainties in the body frame. In cases where several frames are used simultaneously, a good rule of thumb is that measurements and uncertainties will be represented in the "last" frame of the series.

2 Frame Conventions in Geometry, Lie Group, and Manifold Operations

At the core of most coordinate frame usage in GTSAM are geometry and Lie group operations. We explain the geometry and Lie group operations in GTSAM in terms of

Syntax	Input	Output	Identities
$\delta = a.\mathbf{localCoordinates}(b)$	a in g b in g	δ in a	$a.\mathbf{retract}(\delta) == b$ $\mathbf{I}.\mathbf{localCoordinates}($ $a.\mathbf{between}(b)) == \delta$
$b = a.\mathbf{retract}(\delta)$	a in g δ in a	b in g	$a.\mathbf{compose}($ $\mathbf{I}.\mathbf{retract}(\delta)) == b$

Table 2: Coordinate frames for manifold tangent space operations. Here, a, b, and g are manifold elements, δ is a tangent space element, and X is a Lie group type (e.g. Pose2). For the identities column, we assume the elements are also Lie group elements with identity **I**.

the coordinate frame transformations they perform, detailed in Table 1.

The manifold tangent space operations "retract" and "local coordinates" also work in the body frame for Lie group elements. The tangent space coordinates given to "retract" should be in the body frame, not the global frame. Similarly, the tangent space coordinates returned by "local coordinates" will be in the same body frame. This is detailed in Table 2.

3 Frame and Uncertainty Conventions For Built-in Factors

All built-in GTSAM factors follow a consistent coordinate frame convention (though fundamentally how a measurement and its uncertainty are specified depends on the measurement model described by a factor). In all built-in GTSAM factors, the *noise model*, i.e. the measurement uncertainty, should be specified in the coordinate frame of the measurement itself. This is part of a convention in GTSAM that tangent-space quantities (like Gaussian noise models and update delta vectors) are always in the coordinate frame of the element owning the tangent space.

3.1 PriorFactor

A PriorFactor is a simple unary prior. It encodes a direct measurement of the value of a variable x, with the specified mean z and uncertainty, such that $z.\text{between}(x)$ is distributed according to the specified noise model. From this definition and the definition of **between** in Table 1, the measurement itself should be specified in the frame with respect to which x is specified, while the uncertainty is specified in the coordinate frame of the measurement, or equivalently, in frame x.

3.2 BetweenFactor

A BetweenFactor is a measurement on the relative transformation between two variables. A BetweenFactor on variables x and y with measurement z implies that $z.\text{between}(x.\text{between}(y))$

Name	Residual	Variables	Measurement (z)	Measurement Uncertainty
PriorFactor	z.localCoordinates(x)	x in g	Ideal x in g	In z / In x
BetweenFactor	z.localCoordinates(x.between(y))	x in g y in g	Ideal y in x	In z / In y
RangeFactor	x.range$(y) - z$	x in g y in g	Euclidean distance	In z
BearingFactor	z.localCoordinates(x.bearing(y))	x in g y in g	Bearing of y position in frame x	In z
GenericProjection Factor	$K^{-1}\left(P\left(x^{-1}p\right)\right) - z$	x in g p in g	Perspective projection of p in x.	In z
GeneralSFM Factor	$K^{-1}\left(P\left(x^{-1}p\right)\right) - z$	x in g p in g Parameters of K	Perspective projection of p in x.	In z

Table 3: Measurement functions and coordinate frames of factors provided with GT-SAM. To simplify notation, K is a camera calibration function converting pixels to normalized image coordinates, and P is the pinhole projection function.

is distributed according to the specified noise model. This definition, along with that of **between** in Table 1, implies that the measurement is in frame x, i.e. it measures y in x, and that the uncertainty is in the frame of the measurement, or equivalently, in frame y.

3.3 RangeFactor

A RangeFactor measures the Euclidean distance either between two poses, a pose and a point, or two points. The range is a scalar, specified to be distributed according to the specified noise model.

3.4 BearingFactor

A BearingFactor measures the bearing (angle) of the *position* of a pose or point y as observed from a pose x. The orientation of x affects the measurement prediction. Though, if y is a pose, it's orientation does not matter. The noise model specifies the distribution of the bearing, in radians.

3.5 GenericProjectionFactor

A GenericProjectionFactor measures the pixel coordinates of a landmark p projected into a camera x with the calibration function K that converts pixels to normalized image coordinates. The measurement z is specified in real pixel coordinates (thanks to the "uncalibration" function K^{-1} used in the residual). In a GenericProjectionFactor, the calibration is fixed. On the other hand, GeneralSFMFactor allows the calibration parameters to be optimized as variables.

3.6 GeneralSFMFactor

A GeneralSFMFactor is the same as a GenericProjectionFactor except that a GeneralSFMFactor also allows the parameters of the calibration function K to be optimized as variables, instead of having them fixed. A GeneralSFMFactor measures the pixel coordinates of a landmark p projected into a camera x with the calibration function K that converts pixels to normalized image coordinates. The measurement z is specified in real pixel coordinates (thanks to the "uncalibration" function K^{-1} used in the residual).

4 Noise models of prior factors

The simplest way to describe noise models is by an example. Let's take a prior factor on a 3D pose $x \in \mathbb{SE}(3)$, `Pose3` in GTSAM. Let $z \in \mathbb{SE}(3)$ be the expected pose, i.e. the zero-error solution for the prior factor. The *unwhitened error* (the error vector not accounting for the noise model) is

$$h(x) = \log\left(z^{-1}x\right),$$

where $\cdot^{-1}$ is the Lie group inverse and $\log\cdot$ is the logarithm map on $\mathbb{SE}(3)$. The full factor error, including the noise model, is

$$e(x) = \|h(x)\|_{\Sigma}^{2} = h(x)^{\mathsf{T}}\,\Sigma^{-1}h(x).$$

[Skipping details of the derivation for now, for lack of time to get a useful answer out quickly]

The density induced by a noise model on the prior factor is Gaussian in the tangent space about the linearization point. Suppose that the pose is linearized at $\mathring{x} \in \mathbb{SE}(3)$, which we assume is near to z. Let $\delta x \in \mathbb{R}^{6}$ be an update vector in local coordinates (a twist). Then, the factor error in terms of the update vector δx is

$$e(\delta x) = \left\|h\left(\mathring{x}\exp\delta x\right)\right\|_{\Sigma}^{2}$$

We can see why the covariance Σ is in the body frame of x by looking at the linearized error function,

$$\begin{aligned} e(\delta x) &\approx \left\|\log\left(z^{-1}\mathring{x}\exp\delta x\right)\right\|_{\Sigma}^{2} \\ &\approx \left\|\log\left(z^{-1}\mathring{x}\right)+\delta x\right\|_{\Sigma}^{2} \end{aligned}$$

Here we see that the update $\exp\delta x$ from the linear step δx is applied in the body frame of $\mathring{x}$, because of the ordering $\mathring{x}\exp\delta x$. Furthermore, $z^{-1}\mathring{x}$ is a constant term, so we can also see that the covariance Σ is actually applied to the linear update vector δx.

This means that to draw random pose samples, we actually draw random samples of δx with zero mean and covariance Σ, i.e.

$$\delta x \sim \mathcal{N}(0, \Sigma).$$

5 Noise models of between factors

The noise model of a BetweenFactor is a bit more complicated. The unwhitened error is

$$h(x_1, x_2) = \log\left(z^{-1} x_1^{-1} x_2\right),$$

where z is the expected relative pose between x_1 and x_2, i.e. the factor has zero error when $x_1 z = x_2$. If we consider the density on the second pose x_2 induced by holding the first pose x_1 fixed, we can see that the covariance is applied to the linear update in the body frame of the second pose x_2,

$$e(\delta x_2) \approx \left\|\log\left(z^{-1} x_1^{-1} x_2 \exp \delta x_2\right)\right\|_\Sigma^2.$$

If we hold the second pose fixed, the covariance is applied as follows (actually, what frame is it in now??)

$$\begin{aligned} e(\delta x_1) &\approx \left\|\log\left(z^{-1}\left(x_1 \exp \delta x_1\right)^{-1} x_2\right)\right\|_\Sigma^2 \\ &= \left\|\log\left(z^{-1} \exp -\delta x_1 x_1^{-1} x_2\right)\right\|_\Sigma^2 \end{aligned}$$