

loam_velodyne のソースコードに関するノート

松谷研究室

2021/4/26

目次

1	前提	3
1.1	ROS の座標軸と LOAM の座標軸	3
1.2	スイープ	4
1.3	Velodyne 社製の LiDAR の座標系	4
2	BasicScanRegistration クラス	4
2.1	PointLabel 列挙体	4
2.2	RegistrationParams クラス	5
2.3	IMUState 構造体	6
2.4	IMUState::interpolate()	6
2.5	BasicScanRegistration::interpolateIMUStateFor()	6
2.6	BasicScanRegistration::setIMUTransformFor()	7
2.7	BasicScanRegistration::transformToStartIMU()	8
2.8	BasicScanRegistration::projectPointToStartOfSweep()	10
2.9	BasicScanRegistration::updateIMUData()	10
2.10	BasicScanRegistration::setScanBuffersFor()	11
2.11	BasicScanRegistration::setRegionBuffersFor()	12
2.12	BasicScanRegistration::extractFeatures()	13
2.13	BasicScanRegistration::markAsPicked()	14
2.14	BasicScanRegistration::updateIMUTransform()	15
3	ScanRegistration クラス	16
3.1	ScanRegistration::parseParams()	16
3.2	ScanRegistration::setupROS()	16
3.3	ScanRegistration::publishResult()	17
3.4	ScanRegistration::handleIMUMessage()	17
4	MultiScanRegistration クラス	18
4.1	MultiScanRegistration::setupROS()	18

4.2	MultiScanRegistration::handleCloudMessage()	18
4.3	MultiScanRegistration::process()	18
5	BasicLaserOdometry クラス	20
5.1	BasicLaserOdometry::transformToStart()	20
5.2	BasicLaserOdometry::transformToEnd()	21
5.3	BasicLaserOdometry::pluginIMURotation()	23
5.4	BasicLaserOdometry::accumulateRotation()	27
5.5	BasicLaserOdometry::updateIMU()	28
5.6	BasicLaserOdometry::process()	28
6	LaserOdometry クラス	47
6.1	LaserOdometry::setup()	47
6.2	LaserOdometry::laserCloudSharpHandler()	49
6.3	LaserOdometry::laserCloudLessSharpHandler()	49
6.4	LaserOdometry::laserCloudFlatHandler()	49
6.5	LaserOdometry::laserCloudLessFlatHandler()	49
6.6	LaserOdometry::laserCloudFullResHandler()	49
6.7	LaserOdometry::hasNewData()	50
6.8	LaserOdometry::publishResult()	50
7	BasicLaserMapping クラス	53
7.1	BasicLaserMapping::transformAssociateMap()	53
7.2	BasicLaserMapping::transformUpdate()	59
7.3	BasicLaserMapping::pointAssociateToMap()	59
7.4	BasicLaserMapping::pointAssociateToBeMapped()	60
7.5	BasicLaserMapping::transformFullResToMap()	60
7.6	BasicLaserMapping::createDownsizedMap()	60
7.7	BasicLaserMapping::process()	61
7.8	BasicLaserMapping::optimizeTransformToBeMapped()	67
8	LaserMapping クラス	76
8.1	LaserMapping::setup()	76
8.2	LaserMapping::laserCloudCornerLastHandler()	78
8.3	LaserMapping::laserCloudSurfLastHandler()	78
8.4	LaserMapping::laserCloudFullResHandler()	78
8.5	LaserMapping::laserOdometryHandler()	79
8.6	LaserMapping::imuHandler()	79
8.7	LaserMapping::hasNewData()	79
8.8	LaserMapping::publishResult()	79

9	BasicTransformMaintenance クラス	80
9.1	BasicTransformMaintenance::transformAssociateMap()	80
10	TransformMaintenance クラス	81
10.1	TransformMaintenance::setup()	81
10.2	TransformMaintenance::laserOdometryHandler()	83
10.3	TransformMaintenance::odomAftMappedHandler()	83

1 前提

1.1 ROS の座標軸と LOAM の座標軸

LOAM の論文 [1] と実装では, Z 軸が前方向, X 軸が左方向, Y 軸が上方向の右手座標系を扱う. 一方 ROS では, X 軸が前方向, Y 軸が左方向, Z 軸が上方向の右手座標系を扱う. この文書では簡単のため, LOAM の座標軸を X, Y, Z , そして ROS の座標軸を $\tilde{X}, \tilde{Y}, \tilde{Z}$ で表す ($\tilde{X} = Z, \tilde{Y} = X, \tilde{Z} = Y$, あるいは $X = \tilde{Y}, Y = \tilde{Z}, Z = \tilde{X}$). LOAM の座標軸における X, Y, Z 軸周りの回転行列を R_x, R_y, R_z で表す. 一方, ROS の座標軸における $\tilde{X}, \tilde{Y}, \tilde{Z}$ 軸周りの回転行列を $\tilde{R}_x, \tilde{R}_y, \tilde{R}_z$ で表す ($\tilde{R}_x = R_z, \tilde{R}_y = R_x, \tilde{R}_z = R_y$). ROS から渡されるデータ (IMU から取得した加速度と速度, LiDAR から取得した点群の座標など) は ROS の座標軸, LOAM 内で計算されるデータ (地図に含まれる各点の座標など) は LOAM の座標軸で表される. ROS と LOAM とでデータをやり取りする部分では, 座標軸の違いに配慮している.

LOAM の座標軸において, X 軸周りの ϕ の回転, Y 軸周りの θ の回転, Z 軸周りの ψ の回転を表す回転行列を $R_x(\phi), R_y(\theta), R_z(\psi)$ とする. 要素ごと書き下すと次のようになる. $\tilde{R}_x, \tilde{R}_y, \tilde{R}_z$ も同様に定義される.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (1)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2)$$

$$R_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

回転は, X, Y, Z 軸周りの回転角 (ロール角 ϕ , ピッチ角 θ , ヨー角 ψ) で表現でき, オイラー角とよばれる. ROS では, オイラー角の回転順序として $\tilde{X} \rightarrow \tilde{Y} \rightarrow \tilde{Z}$ を採用する一方, LOAM の論文と実装では $Z \rightarrow X \rightarrow Y$ となっている. ROS においてオイラー角が $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ であるとき, 対応する回転行列 \tilde{R} は次のようになる ($\tilde{X}, \tilde{Y}, \tilde{Z}$ の順).

$$\tilde{R} = \tilde{R}_z(\tilde{\psi})\tilde{R}_y(\tilde{\theta})\tilde{R}_x(\tilde{\phi}) = R_y(\tilde{\psi})R_x(\tilde{\theta})R_z(\tilde{\phi}) \quad (4)$$

LOAM においてオイラー角が ϕ, θ, ψ であるとき, 対応する回転行列 R は次のようになる (Z, X, Y の順).

$$R = R_y(\theta)R_x(\phi)R_z(\psi) = \tilde{R}_z(\theta)\tilde{R}_y(\phi)\tilde{R}_x(\psi) \quad (5)$$

ROS から LOAM にオイラー角 $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ が渡されたとき, LOAM 側では (4) 式の右辺を使えば, ROS 側と同じ意味をもつ回転を, LOAM 側でも表現できる. 即ち, Z 軸周りに $\tilde{\phi}$, X 軸周りに $\tilde{\theta}$, Y 軸周りに $\tilde{\psi}$ だけ回転させ

る。一方, LOAM から ROS にオイラー角 ϕ, θ, ψ が渡されたとき, ROS 側では (5) 式の右边を使えば, LOAM 側と同じ意味をもつ回転を, ROS 側でも表現できる。即ち, \tilde{X} 軸周りに ψ , \tilde{Y} 軸周りに ϕ , \tilde{Z} 軸周りに θ だけ回転させる。

1.2 スイープ

LOAM の論文内でスイープは, LiDAR が周囲の環境をキャプチャするのに必要な時間として定義されており, 1 回のスイープの間に, 複数の点群が取得されることを想定している。但し実装では, 1 つの点群が取得される度に 1 回のスイープとして扱っており, 1 回のスイープには 1 つの点群しか含まれない。従って実装では, スイープと点群が一对一に対応しており, 両者が明確に区別されているとは限らない。またスイープの時間間隔と, 点群データの取得間隔は等しい (デフォルトでは 100ms である)。LOAM の論文に倣って, k 番目のスイープ (点群) の開始時刻を t_k と表す。

LOAM では `/multiScanRegistration`, `/laserOdometry`, `/laserMapping`, `/transformMaintenance` の 4 ノードが動作している。`/multiScanRegistration` では, LiDAR センサから $k+1$ 番目の点群を受け取って ($k+1$ 番目のスイープ), 点群に含まれる各点の座標を時刻 t_{k+1} に揃える。LiDAR センサは回転しながら点をスキャンするため, 各点の取得された時刻には微妙な差が生じるが, 全ての点があたかも時刻 t_{k+1} で同時に取得されたかのように補正することで, 後続の処理で利用しやすくなる。開始時刻 t_{k+1} に揃えた $k+1$ 番目の点群を \mathcal{P}_{k+1} とする。

`/laserOdometry` では, 時刻 t_{k+1} に揃えられた k 番目の点群 $\bar{\mathcal{P}}_k$ に, 時刻 t_{k+1} で揃えられた最新の点群 \mathcal{P}_{k+1} をレジストレーションすることで, 時刻 t_{k+1} から次の時刻 t_{k+2} にかけての LiDAR の姿勢変化 T_{k+1}^L (LiDAR 座標系) を計算する。`/laserMapping` では, 時刻 t_{k+1} における LiDAR の姿勢 T_k^W (ワールド座標系) に, `/laserOdometry` で求めた姿勢変化 T_{k+1}^L をつなぎ合わせることで, 時刻 t_{k+2} における LiDAR の姿勢 T_{k+1}^W を得る。この姿勢 T_{k+1}^W は, T_{k+1}^W によってワールド座標系に変換された $k+1$ 番目の点群 $\bar{\mathcal{Q}}_{k+1}$ (時刻 t_{k+2} に揃えられている) を, 地図に含まれる点群 \mathcal{Q}_k (スイープ 1 から k までを含み, 各点はワールド座標系で表現) にレジストレーションすることで改良される。

1.3 Velodyne 社製の LiDAR の座標系

`loam_velodyne` パッケージは, Velodyne 社製の LiDAR に対応している。LiDAR センサの座標系を図 1 に示す。角度 α, ω はそれぞれ方位角 (Azimuth), 仰角 (Elevation) とよばれる。スキャン点の極座標 (R, α, ω) が与えられているとき, 対応する直交座標 $(\tilde{x}, \tilde{y}, \tilde{z})$ は $\tilde{x} = R \cos \omega \sin \alpha$, $\tilde{y} = R \cos \omega \cos \alpha$, $\tilde{z} = R \sin \omega$ のように得られる。角度 α は \tilde{Y} 軸を基準とし, 時計回りに回転させると増加, 反時計回りに回転させると減少する (ω とは反対である)。

2 BasicScanRegistration クラス

点群を受け取って特徴抽出を行うノード `scanRegistration` の本体の実装。

2.1 PointLabel 列挙体

点群内の各点に対して, 4 種類のラベルを与える。

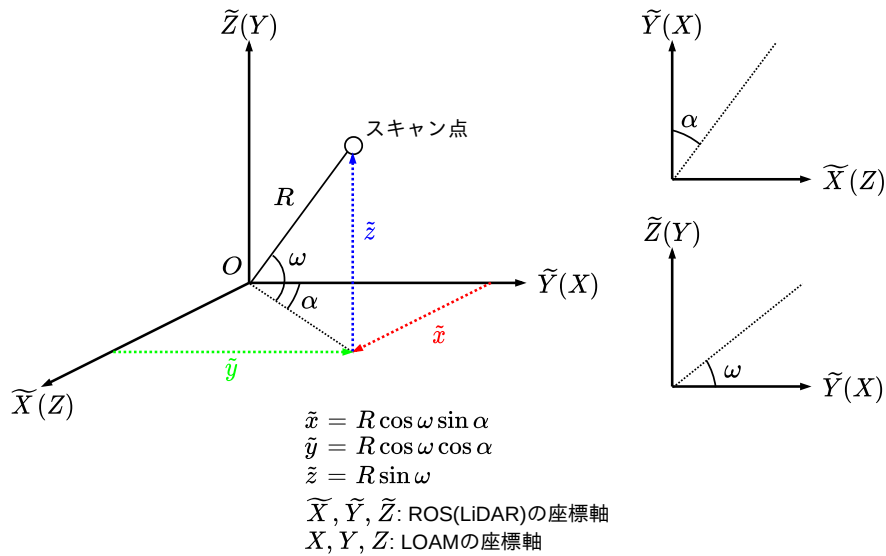


図1 Velodyne 社製の LiDAR の座標系

- CORNER_SHARP: 顕著なコーナー
- CORNER_LESS_SHARP: それほど顕著ではないコーナー
- SURFACE_LESS_FLAT: それほど顕著ではない平面
- SURFACE_FLAT: 顕著な平面

各点の周囲の点を使って曲率 (Curvature) を計算し、曲率の大きさに応じてラベル付けされる。

2.2 RegistrationParams クラス

点群の特徴抽出に関するパラメータを保持する単純なクラス。

- scanPeriod: 点群の取得間隔 (100ms)
- imuHistorySize: IMU からの入力保持するリングバッファのサイズ
- nFeatureRegions: 点群を分割する領域の数で、各領域ごとに特徴が抽出される
- curvatureRegion: 曲率を計算するために使う近傍点の個数
- maxCornerSharp: 特に顕著なコーナー (CORNER_SHARP) として分類する点の最大数
- maxCornerLessSharp: それほど顕著ではないコーナー (CORNER_LESS_SHARP) として分類する点の最大数
- maxSurfaceFlat: 顕著な平面のラベル (SURFACE_FLAT) として分類する点の最大数
- lessFlatFilterSize: 残りの点、即ち、それほど顕著ではない平面 (SURFACE_LESS_FLAT) に属する点の個数を削減するために使われる、ボクセルフィルタのサイズ
- surfaceCurvatureThreshold: コーナーか平面か判別するための曲率の閾値

2.3 IMUState 構造体

IMU のセンサデータを保持するための単純な構造体.

- stamp: タイムスタンプ
- roll: ロール角 (X 軸周りの回転)
- pitch: ピッチ角 (Y 軸周りの回転)
- yaw: ヨー角 (Z 軸周りの回転)
- position: 位置 (グローバル座標系)
- velocity: 速度 (グローバル座標系)
- acceleration: 加速度 (グローバル座標系)

2.4 IMUState::interpolate()

2 つの IMU データを線形補間する. 2 つの IMU データを $(t_0, p_0, v_0), (t_1, p_1, v_1)$ とする (t はタイムスタンプ, p は位置, v は速度) と, 時刻 t における位置 p と速度 v は, 線形補間によって (6) 式, (7) 式のように得られる ($t_0 < t < t_1$).

$$p = \frac{t_1 - t}{t_1 - t_0} p_0 + \frac{t - t_0}{t_1 - t_0} p_1 = \left(1 - \frac{t - t_0}{t_1 - t_0}\right) p_0 + \frac{t - t_0}{t_1 - t_0} p_1 \quad (6)$$

$$v = \frac{t_1 - t}{t_1 - t_0} v_0 + \frac{t - t_0}{t_1 - t_0} v_1 = \left(1 - \frac{t - t_0}{t_1 - t_0}\right) v_0 + \frac{t - t_0}{t_1 - t_0} v_1 \quad (7)$$

ヨー角のみ, 2 つのデータにおけるヨー角の差が 2π を超えるかどうか確認する. 例えば時刻 t_0 から t_1 にかけて, ヨー角が -179° から 2° だけ逆回転して 179° に変化したとする. その中間の時刻 $t = (t_0 + t_1)/2$ における角度は 180° (あるいは -180°) となるべきだが, 上式に代入して計算すると $-179^\circ/2 + 179^\circ/2 = 0^\circ$ となって, 本来とは正反対の方向になってしまう. 角度は 360° 周期であるので, 180° をまたぐと -180° へと不連続に変化し, それによって上記のような不都合が生じる. 両者の角度の差が -180° から 180° の範囲に収まらなければ, 時刻 t_1 における角度に 360° を加減算することで, 正しい値が求められる. この例では, $-179^\circ - 179^\circ = -358^\circ < -180^\circ$ であるから, $179^\circ - 360^\circ = -181^\circ$ のように, 時刻 t_1 における角度から 2π を引いて補正すれば, 線形補間によって $-179^\circ/2 - 181^\circ/2 = -180^\circ$ のように正しい値が得られる. 別の例として, 179° から 2° だけ回転して -179° になったのなら, $179^\circ - (-179^\circ) = 358^\circ > 180^\circ$ であるから, $-179^\circ + 360^\circ = 181^\circ$ のように 2π を足して補正すれば, 線形補間によって $179^\circ/2 + 181^\circ/2 = 180^\circ$ のように正しい値が求まる. ロボットの動作を考えると, ロール角とピッチ角が元々の状態から 180° 以上回転するとは考えられないので, ロール角とピッチ角に関してはこの処理を行わない (ロボットが転倒して上下逆さまになれば 180° 回転したことになる). 但しヨー角はロボットが向いている方向 (Z 軸周りの回転) であり, 方向はいくらでも変化しうるので, この補正処理が必要となる.

2.5 BasicScanRegistration::interpolateIMUStateFor()

最新の点群のタイムスタンプ `_scanTime` を基準とした, 相対時刻 `relTime` における IMU データを線形補間によって算出する (図 2).

`_scanTime` は厳密には、点群内の最初の点が取得されたときの時刻であり、以後スキャン開始時刻とよぶことにする（最後の点が取得されたときの時刻は、スキャン終了時刻とする）。絶対時刻は `_scanTime + relTime` として得られる（これを t とする）。リングバッファ `_imuHistory` に蓄積された IMU データから、時刻 t の前後に得られた、2 つの隣接する IMU データ (`_imuIdx - 1` と `_imuIdx`) を探索する。ソースコードにおける `timeDiff` は図 2 に示すように、時刻 t から、`_imuIdx` 番目の IMU データのタイムスタンプを引いたものである。時刻 t よりも後に届いた IMU データが存在し、`timeDiff` が 0 以下になることが期待されるが、そのような場合には `_imuIdx - 1` 番目と `_imuIdx` 番目の IMU データを使って、時刻 t における IMU データ `outputState` を線形補間によって求める。IMU データが 1 つしかない場合や、時刻 t よりも後に IMU データが届いておらず、`timeDiff` が正となる場合には外挿が必要になる。実装内では外挿は行っておらず、`_imuIdx` 番目の最新の IMU データを、時刻 t における IMU データとしてそのまま使用する。

このメソッドは `setIMUTransformFor()` のなかで呼び出され、計算結果の IMU データは `_imuCur` に格納される。従って `_imuCur` は、時刻 `_scanTime + relTime` における IMU データを表す。

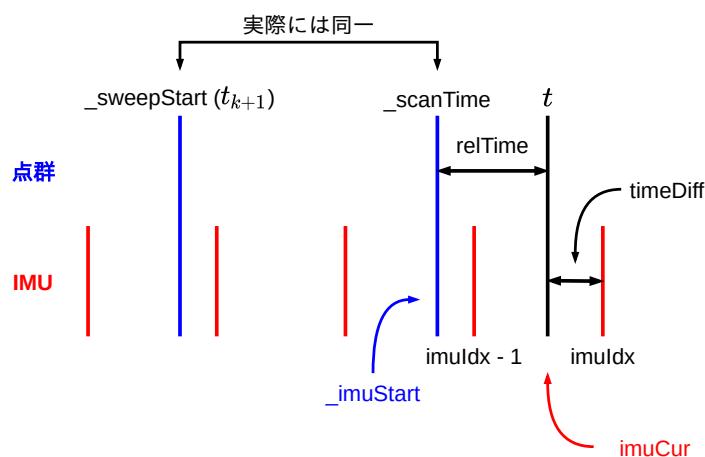


図 2 interpolateIMUStateFor() における IMU データの線形補間

2.6 BasicScanRegistration::setIMUTransformFor()

このメソッドは `projectPointToStartOfSweep()` のなかで最初に呼ばれる。 `interpolateIMUStateFor()` を呼び出して、現在の点群のスキャン開始時刻 `_scanTime` を基準とした、相対時刻 `relTime` における IMU データを線形補間により算出し、`_imuCur` に格納する。続いて、時刻 `_scanTime` から `_scanTime + relTime` までの間、IMU データの速度が一定で、ロボットが定速で移動していた場合の座標と、実際の座標とのずれを計算し、`_imuPositionShift` に格納する (図 4)。ロボットの移動速度は必ずしも一定とは限らないが、LOAM では 1 つの点群を取得する間 (1 回のスイープの間) は定速移動していると仮定される。

`_imuPositionShift` は、`_imuCur.position` から `_imuStart.position + _imuStart.velocity * relSweepTime` を引いたものである。 `relSweepTime` は、現在のスイープ $k + 1$ の開始時刻 `_sweepStart` から、時刻 `_scanTime + relTime` までの経過時間である。但し先述の通り、1 つのスイープは 1 つの点群しか含まないため、現在のスイープ $k + 1$ の開始時刻 `_sweepStart` と、現在の点群 $k + 1$ のスキャン開始時刻 `_scanTime` は同一のものであり、`relSweepTime` は `relTime` と等しくなる。 `_sweepStart` と `_scanTime` はい

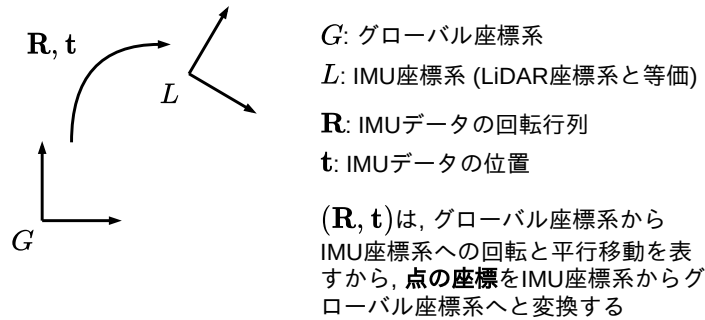


図3 IMU データが表す位置と回転

ずれも, `reset()` のなかで同じ値に設定される。

IMU データにおける位置 (`position` メンバ) はグローバル座標系で表現されるから, `_imuPositionShift` もまたグローバル座標系である。細かく説明すれば, この位置は, グローバル座標系からみた IMU 座標系の位置であるから, 座標系をグローバル座標系から IMU 座標系に変換し, 従って点の座標を IMU 座標系からグローバル座標系へと移す (図 3)。`_imuStart.position + _imuStart.velocity * relTime` は, ロボットが時刻 `_scanTime` 以降, 速度 `_imuStart.velocity` を保って移動していた場合の, 時刻 `_scanTime + relTime` における IMU データの位置である一方, `_imuCur.position` は, 時刻 `_scanTime + relTime` における実際の位置である。両者の差分 `_imuPositionShift` は, 線形補間によって得られた位置と, 実際の位置とのずれを表している。例えば, 時刻 `_scanTime + relTime` で, ロボットからグローバル座標系で x の位置に見える点があったとする。時刻 `_scanTime` から `_scanTime + relTime` までの間に加減速したので x の位置に見えているが, 定速で移動していれば, この点はグローバル座標系で `_imuPositionShift + x` の位置に見えたはずである。この `_imuPositionShift` は `transformToStartIMU()` で使用される。

2.7 BasicScanRegistration::transformToStartIMU()

このメソッドは `projectPointToStartOfSweep()` のなかで, `setIMUTransformFor()` の直後に呼び出されるので, その状況を前提とする。図 5 に示すように, 最新の点群のスキャン開始時刻 (スイープの開始時刻) を `_scanTime` としたとき, 時刻 `_scanTime + relTime` において取得された点の座標 `point` を, 時刻 `_scanTime` における座標に変換する (この変換処理によって, 点 `point` は, あたかも時刻 `_scanTime` で取得されたかのような座標に変換される)。

簡単のため, 点の取得時刻 `_scanTime + relTime` を t , 点の座標 `point` を p とする。 `setIMUTransformFor()` が直前に呼び出されるから, (線形補間によって得られた) 時刻 t における IMU データは `_imuCur` に格納されている。 `_imuCur` における回転 (オイラー角 $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$) は ROS の座標軸で表現される一方, 点の座標 p は LOAM における時刻 t の LiDAR 座標系で表される。(4) 式より, (ROS の座標軸における) 回転 $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ と同じ意味をもつ回転行列 \tilde{R}_c は, LOAM 側では $R_y(\tilde{\psi})R_x(\tilde{\theta})R_z(\tilde{\phi})$ として得られる。

この回転 $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ は, 時刻 t の IMU 座標系がグローバル座標系からどの程度回転しているか, 即ちグローバル座標系から IMU 座標系への座標系の回転を表している (図 5 の座標系 G から C)。言い換えると $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ は, 点の座標を時刻 t の IMU 座標系からグローバル座標系へと移す (図 5 の座標系 C から G)。点の座標 p は時刻 t の LiDAR 座標系 (IMU 座標系と等価) で表されるから, $\tilde{R}_c p$ とすればグローバル座標系に変換できる。

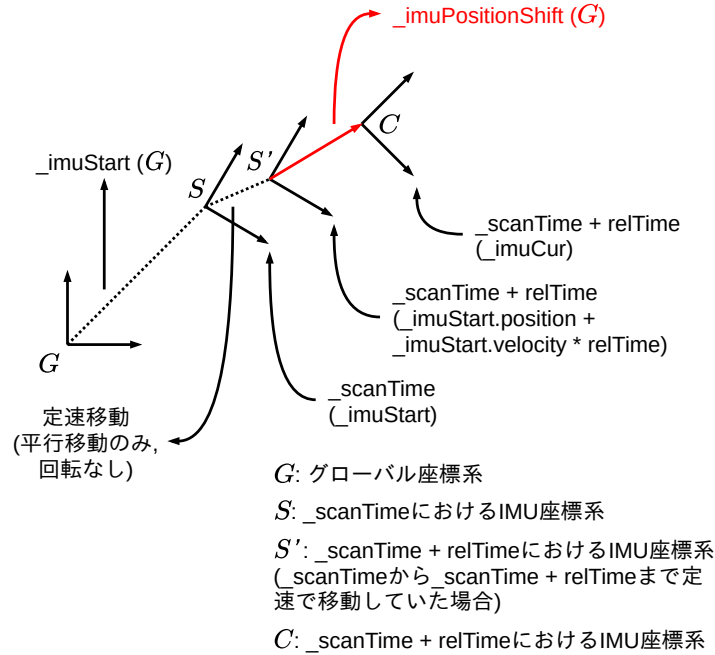


図4 setIMUTransformFor() における処理

\tilde{R}_c は、 Z 軸周りに $\tilde{\phi}$, X 軸周りに $\tilde{\theta}$, Y 軸周りに $\tilde{\psi}$ だけ回転させる。ソースコード上では $\tilde{\phi}$ が $_imuCur.roll$, $\tilde{\theta}$ が $_imuCur.pitch$, $\tilde{\psi}$ が $_imuCur.yaw$ に対応し, rotateZXY() 関数を使って Z, X, Y 軸の順に回転する。

$\tilde{R}_c p$ に $_imuPositionShift$ を加算することで、スキャン開始時刻 $_scanTime$ から時刻 t までの間、ロボットが定速で移動していた場合の座標に補正する。最後に、スキャン開始時刻 $_scanTime$ における回転 $\tilde{\phi}_s, \tilde{\theta}_s, \tilde{\psi}_s$ を使って、グローバル座標系から時刻 $_scanTime$ の LiDAR 座標系に変換する (図5の座標系 G から S または G から S')。先程と同様に、この回転 $\tilde{\phi}_s, \tilde{\theta}_s, \tilde{\psi}_s$ も ROS の座標軸で表される。回転行列 $\tilde{R}_s = R_y(\tilde{\psi}_s)R_x(\tilde{\theta}_s)R_z(\tilde{\phi}_s)$ は、点の座標を時刻 $_scanTime$ の IMU 座標系からグローバル座標系へと移す (図5の座標系 S から G または S' から G)。ここではグローバル座標系から時刻 $_scanTime$ の IMU 座標系へと変換したいから、回転行列の転置 \tilde{R}_s^T を適用する。 $\tilde{R}_s^T = R_z(\tilde{\phi}_s)^T R_x(\tilde{\theta}_s)^T R_y(\tilde{\psi}_s)^T = R_z(-\tilde{\phi}_s)R_x(-\tilde{\theta}_s)R_y(-\tilde{\psi}_s)$ が成立するから、 \tilde{R}_s^T は、 Y 軸周りに $-\tilde{\psi}_s$, X 軸周りに $-\tilde{\theta}_s$, Z 軸周りに $-\tilde{\phi}_s$ だけ回転させる。ソースコード上では $\tilde{\phi}_s$ が $_imuStart.roll$, $\tilde{\theta}_s$ が $_imuStart.pitch$, $\tilde{\psi}_s$ が $_imuStart.yaw$ に対応しており, rotateYXZ() 関数を使って上記の回転処理を実現する。図5に示す通り、時刻 $_scanTime$ における IMU 座標系 S と、時刻 $_scanTime$ から $_scanTime + relTime$ までの間定速で移動していた場合の IMU 座標系 S' は単に平行移動の関係にある。従って、 $_imuPositionShift$ は S, S' 双方の座標系として解釈できるほか、回転行列 \tilde{R}_s はグローバル座標系 G から S , かつ G から S' への座標系の回転を表している。

以上の座標変換をまとめると次のようになる。

1. $_imuCur$ を使って、点の座標 $point$ を時刻 $_scanTime + relTime$ の IMU 座標系 (LiDAR 座標系) から、グローバル座標系に変換する。
2. $_imuPositionShift$ を使って、 $_scanTime$ から $_scanTime + relTime$ までの間、IMU の速度が変化せず、ロボットが定速で移動していた場合の座標へと補正する。

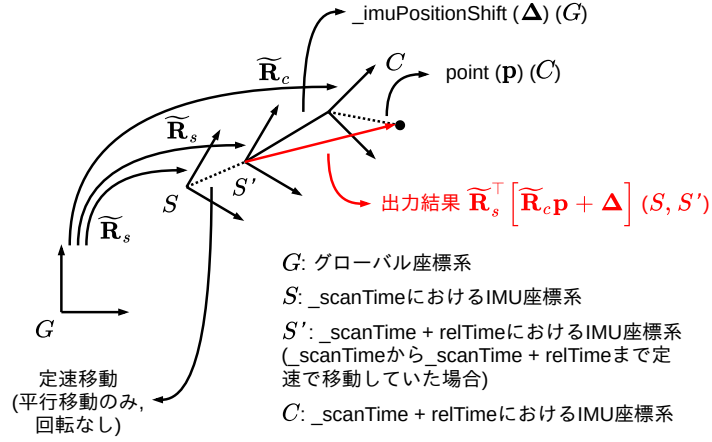


図5 transformToStartIMU() における処理

3. `_imuStart` を使って、グローバル座標系から、時刻 `_scanTime` の IMU 座標系 (LiDAR 座標系) へと変換する。

2.8 BasicScanRegistration::projectPointToStartOfSweep()

最新の点群のスキャン開始時刻 (スイープの開始時刻) を `_scanTime` としたとき、時刻 `_scanTime + relTime` において取得された点の座標 `point` を、時刻 `_scanTime` における座標に変換する。時刻 `_scanTime` での IMU データは既に `_imuStart` に格納されている。 `setIMUTransformFor()` を用いて、時刻 `_scanTime + relTime` における IMU データ `_imuCur` と、定速移動のための補正 `_imuPositionShift` を計算する。続いて、 `transformToStartIMU()` と `_imuCur`, `_imuStart`, `_imuPositionShift` を用いて、時刻 `_scanTime + relTime` における座標 `point` を、時刻 `_scanTime` における座標に変換する (図5)。

2.9 BasicScanRegistration::updateIMUData()

このメソッドは `ScanRegistration` の `handleIMUMessage()` から呼び出される。新しい IMU データ `newState` には、まだ位置 $p(\text{newState.position})$ と速度 $v(\text{newState.velocity})$ が格納されていないため、これらを求める必要がある。求めた後、直近の IMU データを格納するリングバッファ `_imuHistory` の末尾に `newState` を追加する。IMU データの回転 (オイラー角 $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$) は ROS の座標軸である。IMU データの加速度 a は LOAM における IMU 座標系であり、 `handleIMUMessage()` 内の処理によって重力が考慮されている。IMU 座標系からグローバル座標系へと移す回転行列 $\tilde{R} = R_y(\tilde{\psi})R_x(\tilde{\theta})R_z(\tilde{\phi})$ を用いて、加速度 a をグローバル座標系に変換する。ソースコード上では $\tilde{\phi}$ が `newState.roll`, $\tilde{\theta}$ が `newState.pitch`, $\tilde{\psi}$ が `newState.yaw` に対応し、 `rotateZXY()` 関数によって望む回転が行われる。

以前の IMU データ `prevState` の取得時刻 (`prevState.stamp`) を t_0 , 位置 (`prevState.position`) を p_0 , 速度 (`prevState.velocity`) を v_0 , 新しい IMU データ `newState` の取得時刻 (`newState.stamp`) を t_1 とする。新しい位置 p と速度 v は、次のようにして求められる (`timeDiff` は $t_1 - t_0$ に対応する)。

$$p = p_0 + \int_{t_0}^{t_1} (v_0 + \tilde{R}a(t - t_0)) dt = p_0 + v_0(t_1 - t_0) + \frac{1}{2}\tilde{R}a(t_1 - t_0)^2 \quad (8)$$

$$\mathbf{v} = \mathbf{v}_0 + \tilde{\mathbf{R}}\mathbf{a}(t_1 - t_0) \quad (9)$$

先程の `transformToStartIMU()` において、分かりやすさのため、時刻 `_scanTime` を t_0 、時刻 `_scanTime + relTime` を t として、 t_0 における IMU データ `_imuStart` の位置と速度を $\mathbf{p}_0, \mathbf{v}_0$ 、 t における IMU データ `_imuCur` の位置と速度、加速度を $\mathbf{p}, \mathbf{v}, \mathbf{a}$ とおく。 \mathbf{p} と \mathbf{v} は、 $\mathbf{p}_0, \mathbf{v}_0, \mathbf{a}$ から上式のように求められているとする。このとき `_imuPositionShift` は以下のようになり、定速移動による成分 $\mathbf{v}_0(t - t_0)$ は除去され、加速度による成分だけが残っている。従って、加速度が 0(定速移動) であれば `_imuPositionShift` は 0 となるほか、加速度が大きいほど `_imuPositionShift` による補正も強くかかる。

$$\mathbf{p} - (\mathbf{p}_0 + \mathbf{v}_0(t - t_0)) = \left(\mathbf{p}_0 + \mathbf{v}_0(t - t_0) + \frac{1}{2}\tilde{\mathbf{R}}\mathbf{a}(t - t_0)^2 \right) - (\mathbf{p}_0 + \mathbf{v}_0(t - t_0)) = \frac{1}{2}\tilde{\mathbf{R}}\mathbf{a}(t - t_0)^2$$

2.10 BasicScanRegistration::setScanBuffersFor()

開始インデックス `startIdx` および終了インデックス `endIdx` で指定される、点群 `_laserCloud` 内の 1 つのリング (仰角が等しい点のセット) が与えられたとき、信頼性の低い点を除去する (`_scanNeighborPicked` の対応する要素を 1 にする)。各リングは最低でも $2 * \text{curvatureRegion} + 1$ 個の点を含む。 `startIdx + curvatureRegion` から `endIdx - curvatureRegion - 1` までの各点について順に走査し、次の処理を行う。

現在の点 `point` と、その前後の点 `previousPoint`, `nextPoint` を $\mathbf{p}_i, \mathbf{p}_{i-1}, \mathbf{p}_{i+1}$ と表す。 \mathbf{p}_i と \mathbf{p}_{i+1} との距離 $\|\mathbf{p}_i - \mathbf{p}_{i+1}\|$ (`diffNext` の平方根) が約 30cm 以上であれば、2 つの点がオクルージョンの境界にあって、信頼できない可能性がある。このような場合、本当に信頼できないかどうかを次のように判定する。LiDAR から \mathbf{p}_i および \mathbf{p}_{i+1} までの距離 $\|\mathbf{p}_i\|, \|\mathbf{p}_{i+1}\|$ (`depth1, depth2`) を計算する。 $\|\mathbf{p}_i\| > \|\mathbf{p}_{i+1}\|$ であれば (\mathbf{p}_i の方が LiDAR から離れている)、距離の比率 $\|\mathbf{p}_{i+1}\| / \|\mathbf{p}_i\| < 1$ を使って正規化した点 $\mathbf{p}_i (\|\mathbf{p}_{i+1}\| / \|\mathbf{p}_i\|)$ (LiDAR からの距離は \mathbf{p}_i に等しいが、 \mathbf{p}_{i+1} と同じ方向にある点) と、 \mathbf{p}_{i+1} との距離 d_w (`weighted_distance`) を計算する。この d_w は次のように、LiDAR から 2 つの点 $\mathbf{p}_i, \mathbf{p}_{i+1}$ までの方向の差 θ と関連している。

$$\begin{aligned} d_w &= \left(\left\| \mathbf{p}_{i+1} - \frac{\|\mathbf{p}_{i+1}\|}{\|\mathbf{p}_i\|} \mathbf{p}_i \right\|^2 \right)^{\frac{1}{2}} \\ &= \left(\mathbf{p}_{i+1}^\top \mathbf{p}_{i+1} + \left(\frac{\|\mathbf{p}_{i+1}\|}{\|\mathbf{p}_i\|} \right)^2 \mathbf{p}_i^\top \mathbf{p}_i - 2 \frac{\|\mathbf{p}_{i+1}\|}{\|\mathbf{p}_i\|} \mathbf{p}_{i+1}^\top \mathbf{p}_i \right)^{\frac{1}{2}} \\ &= \left(\|\mathbf{p}_{i+1}\|^2 + \left(\frac{\|\mathbf{p}_{i+1}\|}{\|\mathbf{p}_i\|} \right)^2 \|\mathbf{p}_i\|^2 - 2 \frac{\|\mathbf{p}_{i+1}\|}{\|\mathbf{p}_i\|} \mathbf{p}_{i+1}^\top \mathbf{p}_i \right)^{\frac{1}{2}} \\ &= \left(2 \|\mathbf{p}_{i+1}\|^2 - 2 \frac{\|\mathbf{p}_{i+1}\|}{\|\mathbf{p}_i\|} \mathbf{p}_{i+1}^\top \mathbf{p}_i \right)^{\frac{1}{2}} \\ &= \left(\|\mathbf{p}_{i+1}\|^2 \left(2 - 2 \frac{\mathbf{p}_{i+1}^\top \mathbf{p}_i}{\|\mathbf{p}_{i+1}\| \|\mathbf{p}_i\|} \right) \right)^{\frac{1}{2}} \end{aligned}$$

ゆえ

$$\frac{d_w}{\|\mathbf{p}_{i+1}\|} = (2 - 2 \cos \theta)^{\frac{1}{2}} = \left(4 \cdot \frac{1 + \cos \theta}{2} \right)^{\frac{1}{2}} = \left(4 \sin^2 \frac{\theta}{2} \right)^{\frac{1}{2}} = 2 \sin \frac{\theta}{2}$$

が成立する ($0 \leq \theta \leq 2\pi$)。 d_w は、2 つの点の方向が互いに近ければ (θ が小さい) 0 に近づき、方向が正反対であれば ($\theta = 2\pi$) 最大となる。 $d_w < 0.1$ であれば、LiDAR からみて 2 つの点の方向はそれほど変わらない

いのに、互いに距離が離れている、即ちオクルージョンの境界である可能性が高い (LOAM の論文の図 4 を参照) から、 p_i と p_{i+1} は信頼できない点である。従って、現在の点 $point$ とそれ以前の $curvatureRegion$ 個の点を使わないように、これらの点に対応する $scanNeighborPicked$ の要素にフラグをセットする。上記は $\|p_i\| > \|p_{i+1}\|$ の場合であるが、 $\|p_i\| \leq \|p_{i+1}\|$ の場合も同様の処理を行う ($d_w < 0.1$ であれば、次の点 $nextPoint$ とそれ以降の $curvatureRegion$ 個の点を使わないように、フラグをセットする)。 $scanNeighborPicked$ は、 $startIdx$ から $endIdx$ までのインデックスをもつ点について割り当てられるフラグである。フラグが 1 であればその点は除外されたことを示し、後続の処理 (点に対するラベル付け) などで無視される。

最後に、直前の点 p_{i-1} から現在の点 p_i までの距離 $\|p_i - p_{i-1}\|$ ($diffPrevious$ の平方根) を計算する。隣接する点までの距離 $\|p_i - p_{i-1}\|$ と $\|p_i - p_{i+1}\|$ が両方とも、LiDAR から現在の点 p_i までの距離 $\|p_i\|$ (dis の平方根) の 1% 以上であれば、現在の点は周囲の点から孤立している可能性が高いので除外する。

2.11 BasicScanRegistration::setRegionBuffersFor()

開始インデックス $startIdx$ および終了インデックス $endIdx$ で指定される、点群 $laserCloud$ 内の小領域が与えられたとき、小領域内の各点に対して曲率を計算し、曲率について降順に点をソートする。小領域内の点は、いずれも同一のリング (仰角が等しい点のセット) に属している。 $startIdx$ から $endIdx$ までの各点について順に走査し、次の処理を行う。

現在の点 $point$ について、その周囲 $curvatureRegion$ 個の点 ($i - curvatureRegion$ から $i + curvatureRegion$ まで) を使って曲率を計算し、 $regionCurvature$ に曲率を格納する ($regionSortIndices$ にはソートのためのインデックスを格納する)。周囲の点のセットを $S = \{p_j\}$ 、現在の点を p_i とすると、現在の点に対する曲率 c_i は次のように表される。

$$c_i = \frac{1}{|S| \|p_i\|} \sum_{p_j \in S} \|p_i - p_j\| \quad (10)$$

曲率は、計算に用いた周囲の点の個数 $|S|$ と、LiDAR から現在の点までの距離 $\|p_i\|$ を使って正規化するのが正しいが、実装内では正規化が省略されている。小領域内の点に対して曲率を求めたら、曲率が大きい順に点をソートする。各点は、点群のインデックスと、小領域のインデックスの両方もち、2 つのインデックスは丁度 $startIdx$ だけずれている (例えば $startIdx$ と $regionIdx$)。前者は $laserCloud$ (全てのリングと全ての点を含む)、後者は $regionCurvature$ や $regionSortIndices$ 、 $regionLabel$ 内の要素を指し示す際に用いる。ソートによって $regionSortIndices$ の先頭要素には、小領域内で曲率が最大となる点を指す、点群のインデックスが格納される。最大の曲率は $regionCurvature[regionSortIndices[0] - startIdx]$ 、最大の曲率をもつ点の座標は $laserCloud[regionSortIndices[0]]$ として得られる。

ソースコードの途中に出現する

```

1   float pointWeight = -2 * _config.curvatureRegion;
2   for (size_t i = startIdx, regionIdx = 0; i <= endIdx; i++, regionIdx++) {
3       float diffX = pointWeight * _laserCloud[i].x;
4       float diffY = pointWeight * _laserCloud[i].y;
5       float diffZ = pointWeight * _laserCloud[i].z;
6       for (int j = 1; j <= _config.curvatureRegion; j++) {
7           diffX += _laserCloud[i + j].x + _laserCloud[i - j].x;
8           diffY += _laserCloud[i + j].y + _laserCloud[i - j].y;
9           diffZ += _laserCloud[i + j].z + _laserCloud[i - j].z;

```

```

10     }
11     _regionCurvature[regionIdx] = diffX * diffX + diffY * diffY + diffZ * diffZ;
12     _regionSortIndices[regionIdx] = i;
13 }

```

の行は、以下のように直したほうが分かりやすい。

```

1   for (size_t i = startIdx, regionIdx = 0; i <= endIdx; i++, regionIdx++) {
2       float diffX = 0.f;
3       float diffY = 0.f;
4       float diffZ = 0.f;
5       for (int j = 1; j <= _config.curvatureRegion; j++) {
6           diffX += (_laserCloud[i + j].x - _laserCloud[i].x) +
7                   (_laserCloud[i - j].x - _laserCloud[i].x);
8           diffY += (_laserCloud[i + j].y - _laserCloud[i].y) +
9                   (_laserCloud[i - j].y - _laserCloud[i].y);
10          diffZ += (_laserCloud[i + j].z - _laserCloud[i].z) +
11                  (_laserCloud[i - j].z - _laserCloud[i].z);
12      }
13      _regionCurvature[regionIdx] = diffX * diffX + diffY * diffY + diffZ * diffZ;
14      _regionSortIndices[regionIdx] = i;
15  }

```

2.12 BasicScanRegistration::extractFeatures()

現在の点群 `_laserCloud` の各点を、顕著なコーナー (CORNER_SHARP)、それほど顕著ではないコーナー (CORNER_LESS_SHARP)、それほど顕著ではない平面 (SURFACE_LESS_FLAT)、顕著な平面 (SURFACE_FLAT) の 4 種類に分類する。点群 `_laserCloud` は図 6 に示すように、各リングに対する点群が連結されたものであり、リングの始点と終点に対応するインデックスは `_scanIndices` で管理されている。`_laserCloud` の内の各リング (`scanStartIdx` から `scanEndIdx` までのインデックス) に対して、次の処理を行う。

現在のリングが $2 * \text{curvatureRegion} + 1$ 個未満の点しか含まなければ、曲率を計算できないためリング自体を除外する。続いて `setScanBuffersFor()` を呼び出し、信頼できない点をリングから除外する。リングを `nFeatureRegions` 個の小領域に分割し、各小領域 (`_laserCloud` 内の `sp` から `ep` までのインデックス) について次の処理を行う。

`setRegionBuffersFor()` を呼び出し、小領域内の各点の曲率を求めて、曲率について降順に点をソートする。小領域内の全ての点を、それほど顕著ではない平面として暫定的にラベル付けする。曲率最大の点を最大 `maxCornerLessSharp` 個選んで、それほど顕著ではないコーナーのラベルを割り当て、点群 `_cornerPointsLessSharp` に追加する。それほど顕著ではないコーナーとして分類された点のなかで、曲率最大の点 `maxCornerSharp` 個に、顕著なコーナーのラベルを割り当て、点群 `_cornerPointsSharp` に追加する。従って `_cornerPointsLessSharp` は `_cornerPointsSharp` を含む。ある点がコーナーとして選ばれたら、その近傍の点がコーナーとして選ばれないように、`markAsPicked()` を呼び出して `_scanNeighborPicked` を更新する。また点がコーナーとして選ばれるためには、対応する `_scanNeighborPicked` の要素が 0 で (その点がまだ除外されていない)、かつ曲率が閾値 `surfaceCurvatureThreshold` を上回る必要がある。

曲率最小の点を最大 `maxSurfaceFlat` 個選んで、顕著な平面のラベルを割り当て、点群 `_surfacePointsFlat` に追加する。ある点が平面として選ばれたら、その近傍の点が平面として選ばれないように、`markAsPicked()` を呼び出して `_scanNeighborPicked` を更新する。また点が平面として選ばれるためには、

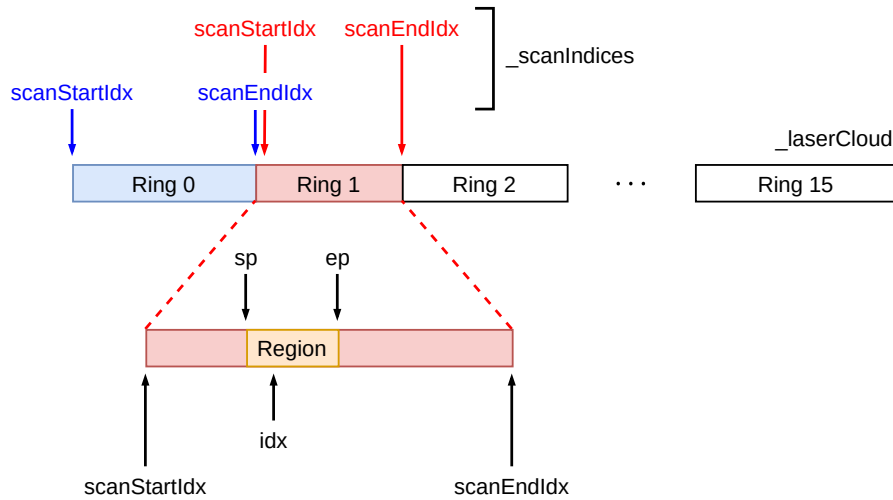


図6 extractFeatures() 内で用いるインデックス

対応する `_scanNeighborPicked` の要素が 0 で (その点がまだ除外されていない), かつ曲率が閾値 `surfaceCurvatureThreshold` 未滿となる必要がある. 上記の処理でラベルが割り振られなかった点と, 顕著な平面としてラベル付けされた点は, いずれも点群 `surfPointsLessFlatScan` に格納する. これをボクセルフィルタでダウンサンプリングしたものを `_surfacePointsLessFlat` に追加する.

途中に出現する

```

1  sp = ((scanStartIdx + _config.curvatureRegion) * (_config.nFeatureRegions - j)
2      + (scanEndIdx - _config.curvatureRegion) * j) / _config.nFeatureRegions;
3  ep = ((scanStartIdx + _config.curvatureRegion) * (_config.nFeatureRegions - 1 - j)
4      + (scanEndIdx - _config.curvatureRegion) * (j + 1)) / _config.nFeatureRegions - 1;

```

の行は, 以下のように直したほうが分かりやすい.

```

1  sp = (scanStartIdx + _config.curvatureRegion) +
2      (scanEndIdx - scanStartIdx - 2 * _config.curvatureRegion) /
3      _config.nFeatureRegions * j;
4  ep = (scanStartIdx + _config.curvatureRegion) +
5      (scanEndIdx - scanStartIdx - 2 * _config.curvatureRegion) /
6      _config.nFeatureRegions * (j + 1) - 1

```

`scanStartIdx + _config.curvatureRegion` は走査を開始するインデックス, `scanEndIdx - scanStartIdx - 2 * _config.curvatureRegion` は現在のリングのなかで曲率を計算可能な点の個数である (両端の `curvatureRegion` 個が除かれている). 点の個数を `_config.nFeatureRegions` で割れば, リング内の各小領域に割り当てる点の個数が定まる. `j` はリング内の小領域のインデックス, `sp` と `ep` は `j` 番目の小領域に含まれる点の範囲を意味する.

2.13 BasicScanRegistration::markAsPicked()

点群 `_laserCloud` 内の指定されたインデックス `cloudIdx` の点と, その周囲の `curvatureRegion` 個の点を除外し (`_scanNeighborPicked` の対応する要素を 1 にする), コーナーや平面の割り当てに使用されないようにする. 但し, 指定された点とその周囲の点が 20cm 以上離れているなら除外しない.

2.14 BasicScanRegistration::updateIMUTransform()

`_imuTrans` は 3 次元ベクトルが 4 要素並んだ配列である。 `_imuTrans` の 0 番目と 1 番目の要素には、 `_imuStart` と `_imuCur` の回転成分 (オイラー角) を格納する。 `_imuStart` は、最新の点群のスキャン開始時刻 (現在のスイープの開始時刻) `_scanTime` における IMU データである。 `_imuCur` は、 `MultiScanRegistration` の `process()` を通じて `projectPointToStartOfSweep()` が呼ばれる度に更新される。 `updateIMUTransform()` が呼ばれたとき、最後の `projectPointToStartOfSweep()` の呼び出しによって更新された `_imuCur` が、 `_imuTrans` に格納される。 `MultiScanRegistration` の `process()` をみると、点群内の各点に対して順番に `projectPointToStartOfSweep()` が呼ばれるので、 `updateIMUTransform()` が呼ばれた時点での `_imuCur` には、点群内の最後の点に対応する `_imuCur` が格納されている。 `_imuCur` は、点群のスキャン開始時刻 `_scanTime` から `_scanPeriod` (点群の取得間隔) だけ経過した、時刻 `_scanTime + _scanPeriod` における IMU データとみなせる (`_scanTime + _scanPeriod` は、次の点群のスキャン開始時刻である)。 IMU データの回転成分はオイラー角によって表現され、ROS の座標軸における、グローバル座標系から IMU 座標系への座標系の回転を意味する。

`_imuPositionShift` は図 4 または図 5 に示す通り、時刻 `_scanTime` から `_scanTime + _scanPeriod` までの間定速で移動していた場合の座標と、時刻 `_scanTime + _scanPeriod` における実際の座標との差分を表す (図 4 と図 5 において、 `relTime` は `_scanPeriod` となる)。 `_imuPositionShift` はグローバル座標系で表現される。 `_imuStart` の回転 (オイラー角 $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$) から作成した回転行列 $\tilde{R} = R_y(\tilde{\psi})R_x(\tilde{\theta})R_z(\tilde{\phi})$ は、(LOAM の座標軸において) グローバル座標系から、時刻 `_scanTime` の IMU 座標系への座標系の回転を表すので、点の座標を時刻 `_scanTime` の IMU 座標系からグローバル座標系へと移す (図 4, 図 5 の座標系 S から G , あるいは S' から G)。 いまは、 `_imuPositionShift` をグローバル座標系から IMU 座標系へと変換したいので (図 4, 図 5 の座標系 G から S , あるいは G から S')、回転行列の転置 \tilde{R}^T を適用する。 転置を計算すると $\tilde{R}^T = R_z(\tilde{\phi})^T R_x(\tilde{\theta})^T R_y(\tilde{\psi})^T = R_z(-\tilde{\phi})R_x(-\tilde{\theta})R_y(-\tilde{\psi})$ であるから、 `rotateXYZ()` を使用して所望の回転操作を行う (`_imuStart.roll` は $\tilde{\phi}$, `_imuStart.pitch` は $\tilde{\theta}$, `_imuStart.yaw` は $\tilde{\psi}$ である)。 \tilde{R}^T を利用して IMU 座標系に変換した `_imuPositionShift` を、 `_imuTrans` の 2 番目の要素に格納する。

`imuVelocityFromStart` は、 `_imuCur.velocity` と `_imuStart.velocity` との差であり、時刻 `_scanTime` から `_scanTime + _scanPeriod` にかけての速度変化をグローバル座標系で表す。 グローバル座標系から、時刻 `_scanTime` の IMU 座標系へ変換するため、上記と同様に回転行列の転置 \tilde{R}^T を適用し、 `_imuTrans` の 3 番目の要素に格納する。

図 4 または図 5 に示す通り、時刻 `_scanTime + _scanPeriod` における IMU 座標系 S と、時刻 `_scanTime` から `_scanTime + _scanPeriod` までの間定速で移動していた場合の IMU 座標系 S' は、互いに平行移動の関係にある。 従って回転行列 \tilde{R}^T は、点の座標をグローバル座標系 G から S, S' 双方の座標系へと回転させるから、 `_imuTrans` の 2 番目と 3 番目の要素は、 S と S' 双方の座標系として解釈できる。

`_imuTrans` が保持する要素は次のようにまとめられる。

- `_imuStart(roll, pitch, yaw)`: ROS の座標軸における、グローバル座標系から、時刻 `_scanTime` の IMU 座標系 (図 4, 図 5 の S) への座標系の回転。 あるいは、グローバル座標系から、時刻 `_scanTime` から `_scanTime + _scanPeriod` までの間定速移動していた場合の IMU 座標系 (図 4, 図 5 の S') への回転。

- `_imuCur(roll, pitch, yaw)`: ROS の座標軸における, グローバル座標系から, 時刻 `_scanTime + _scanPeriod` (次の点群のスキャン開始時刻) の IMU 座標系 (図 4, 図 5 の C) への座標系の回転
- `imuShiftFromStart`: IMU (LiDAR) 座標系 (図 4 または図 5 の S, S') における, `_scanTime` から `_scanTime + _scanPeriod` にかけての速度変化を考慮するための補正
- `imuVelocityFromStart`: IMU (LiDAR) 座標系 (図 4 または図 5 の S, S') における, `_scanTime` から `_scanTime + _scanPeriod` にかけての速度変化

3 ScanRegistration クラス

点群を受け取って特徴抽出を行うノード `scanRegistration` のうち, ROS に関わる処理のみを取り出したクラス.

3.1 `ScanRegistration::parseParams()`

ノードの設定を ROS から受け取って, 点群の特徴抽出に関するパラメータを設定する.

3.2 `ScanRegistration::setupROS()`

ノードが Subscribe あるいは Publish するトピックを設定する. Subscribe するのは次の 2 つのトピックである.

- `/multi_scan_points` トピック (`sensor_msgs::PointCloud2` 型): Velodyne 社製の LiDAR センサから送信された点群. このトピックの設定は `MultiScanRegistration::setupROS()` の中で行われる.
- `/imu/data` トピック (`sensor_msgs::Imu` 型): IMU データ (速度と加速度).

Publish するのは次の 6 つのトピックである.

- `/velodyne_cloud_2` トピック (`sensor_msgs::PointCloud2` 型): 整形された点群. NaN や無限大などの不正な座標をもつ点が除去されている. それ以外の点については, スキャン ID および時刻のオフセットが, `Intensity` 値として格納されている. `Intensity` は反射強度を意味するので, このフィールドの正しい使い方ではない. スキャン ID とは, 0 から 15 までのリングの番号を指し, 1 つのリングに属する全ての点は, 同一の仰角 (Vertical Angle) をもつ. LiDAR からみて水平であれば仰角は 0° である. VLP-16 センサの場合, -15° から 15° まで 2° 刻みで仰角が変化するので, リングは合計 16 つある. 時刻のオフセットとは, 点群内の最初の点が取得されてから, その点が取得されるまでの経過時間である.
- `/laser_cloud_sharp` トピック (`sensor_msgs::PointCloud2` 型): コーナーとして分類された点群. `extractFeatures()` 内で得られた `_cornerPointsSharp`.
- `/laser_cloud_less_sharp` トピック (`sensor_msgs::PointCloud2` 型):それほど顕著ではないコーナーとして分類された点群. `extractFeatures()` 内で得られた `_cornerPointsLessSharp`.
- `/laser_cloud_flat` トピック (`sensor_msgs::PointCloud2` 型): 平面として分類された点群. `extractFeatures()` 内で得られた `_surfacePointsFlat`.
- `/laser_cloud_less_flat` トピック (`sensor_msgs::PointCloud2` 型): それほど顕著ではない平面

として分類された点群. `extractFeatures()` 内で得られた `_surfacePointsLessFlat`(数が多いのでボクセルフィルタによりダウンサンプリングされている).

- `/imu_trans` トピック (`sensor_msgs::PointCloud2` 型): `updateIMUTransform()` 内で計算された IMU に関連するデータ. データ型は点群 (`PointCloud2`) であるが, 単に座標の配列の格納に便利だから使われているだけで, 点群とは関係ない.

3.3 `ScanRegistration::publishResult()`

6つのトピック (点群と IMU データ) を Publish する.

3.4 `ScanRegistration::handleIMUMessage()`

ROS から IMU データを受け取った際に呼び出されるコールバックである. 重力の影響を考慮して加速度を補正し, 更に ROS の座標軸を LOAM の座標軸に変換する. その後 `BasicScanRegistration` の `updateIMUData()` を呼び出して, IMU に関する状態を更新する. ROS の座標軸において, 重力加速度のベクトルはグローバル座標系で $\mathbf{g} = [0, 0, -9.81]^\top$ で表される. IMU データのオイラー角 (ロール角 $\tilde{\phi}$, ピッチ角 $\tilde{\theta}$, ヨー角 $\tilde{\psi}$) を使い, 重力のベクトル \mathbf{g} をグローバル座標系から IMU 座標系へと変換する. オイラー角から作成した回転行列 $\tilde{\mathbf{R}} = \tilde{\mathbf{R}}_z(\tilde{\psi})\tilde{\mathbf{R}}_y(\tilde{\theta})\tilde{\mathbf{R}}_x(\tilde{\phi})$ は, 点の座標を IMU 座標系からグローバル座標系へと移すが, その逆を行いたいので, 回転行列の転置 $\tilde{\mathbf{R}}^\top$ を \mathbf{g} に適用する.

$$\begin{aligned}
\tilde{\mathbf{R}}^\top \mathbf{g} &= \left(\tilde{\mathbf{R}}_z(\tilde{\psi})\tilde{\mathbf{R}}_y(\tilde{\theta})\tilde{\mathbf{R}}_x(\tilde{\phi}) \right)^\top [0, 0, -9.81]^\top \\
&= \tilde{\mathbf{R}}_x(\tilde{\phi})^\top \tilde{\mathbf{R}}_y(\tilde{\theta})^\top \tilde{\mathbf{R}}_z(\tilde{\psi})^\top [0, 0, -9.81]^\top \\
&= \tilde{\mathbf{R}}_x(\tilde{\phi})^\top \tilde{\mathbf{R}}_y(\tilde{\theta})^\top \begin{bmatrix} \cos \tilde{\psi} & \sin \tilde{\psi} & 0 \\ -\sin \tilde{\psi} & \cos \tilde{\psi} & 0 \\ 0 & 0 & 1 \end{bmatrix} [0, 0, -9.81]^\top \\
&= \tilde{\mathbf{R}}_x(\tilde{\phi})^\top \tilde{\mathbf{R}}_y(\tilde{\theta})^\top [0, 0, -9.81]^\top \\
&= \tilde{\mathbf{R}}_x(\tilde{\phi})^\top \begin{bmatrix} \cos \tilde{\theta} & 0 & -\sin \tilde{\theta} \\ 0 & 1 & 0 \\ \sin \tilde{\theta} & 0 & \cos \tilde{\theta} \end{bmatrix} [0, 0, -9.81]^\top \\
&= \tilde{\mathbf{R}}_x(\tilde{\phi})^\top \begin{bmatrix} 9.81 \sin \tilde{\theta} \\ 0 \\ -9.81 \cos \tilde{\theta} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \tilde{\phi} & \sin \tilde{\phi} \\ 0 & -\sin \tilde{\phi} & \cos \tilde{\phi} \end{bmatrix} \begin{bmatrix} 9.81 \sin \tilde{\theta} \\ 0 \\ -9.81 \cos \tilde{\theta} \end{bmatrix} = \begin{bmatrix} 9.81 \sin \tilde{\theta} \\ -9.81 \sin \tilde{\phi} \cos \tilde{\theta} \\ -9.81 \cos \tilde{\phi} \cos \tilde{\theta} \end{bmatrix} \tag{11}
\end{aligned}$$

$\tilde{\mathbf{R}}^\top \mathbf{g}$ を IMU の加速度 (`linear_acceleration`) に加えることで, 重力の影響を考慮した加速度が得られる. ソースコード内では $\tilde{\phi}$ が roll, $\tilde{\theta}$ が pitch, $\tilde{\psi}$ が yaw に対応する. 重力を加えた加速度ベクトルは ROS の座標軸であり, LOAM の座標軸に交換 (\tilde{Y} を X に, \tilde{Z} を Y に, \tilde{X} を Z に) したうえで `newState` の `acceleration` メンバに格納する. `newState` の roll, pitch, yaw メンバは依然として ROS の座標軸におけるオイラー角を表す. `updateIMUData()` の呼び出しによって, `newState` の `position`, `velocity` メンバが埋められる.

4 MultiScanRegistration クラス

点群を受け取って特徴抽出を行うノード `scanRegistration` のうち、点群に対する前処理のみを取り出したクラス。

4.1 MultiScanRegistration::setupROS()

ノードの設定を ROS から受け取って、使用する LiDAR のパラメータ (最小の仰角, 最大の仰角, リングの個数) を設定する。例えば Velodyne 社製の VLP-16 であれば, 最小の仰角は -15° , 最大の仰角は 15° であり (LiDAR に対して水平な角度が 0°), この間を 2° 刻みで変化させながらスキャンするので, リングの個数は 16 となる。LiDAR のパラメータは MultiScanMapper クラスのインスタンス `_scanMapper` が保持しており, 指定された仰角に対応するリング (例えば 0 から 15) は, MultiScanMapper の `getRingForAngle()` から得られる。Velodyne 社製の VLP-16, HDL-32, HDL-64E に関するパラメータは既に用意されている。ノードで `/multi_scan_points` トピックを Subscribe するための設定も行う。

4.2 MultiScanRegistration::handleCloudMessage()

ROS から点群を受け取った際に呼び出されるコールバックである。受け取った点群メッセージ (`sensor_msgs::PointCloud2` 型) を, PCL(Point Cloud Library) における点群データ (`pcl::PointCloud` 型) に変換したうえで, `process()` を呼び出す。

4.3 MultiScanRegistration::process()

ROS から点群を受け取る度に, `handleCloudMessage()` から呼び出される。最新の点群 `laserCloudIn` において, 最初と最後の点の方向 (図 1 参照) を `startOri` と `endOri` に格納する。点の \tilde{Y}, \tilde{Z} 座標から `atan2()` によって求めた角度 θ は, 図 1 の α を使って $\pi/2 - \alpha$ と表せる。Velodyne 社の LiDAR は右回転しながらスキャンするため, 方位角 α は最初の 0 番目の点から最後の `cloudSize - 1` 番目の点に向けて増加, 従って `atan2()` によって求めた角度 $\theta = \pi/2 - \alpha$ は次第に減少していく。角度 θ が最初から最後に向けて増加していくように, `atan2()` で求めた値の符号を反転させ, `startOri` と `endOri` に格納している。以後簡単のため, `startOri` と `endOri` をそれぞれ θ_S, θ_E で表す。

最初の if 文では, $\pi \leq \theta_E - \theta_S \leq 3\pi$ が成立するように, θ_E に 2π を加減して調節する。即ちこのメソッドでは, 少なくとも半周分, 最大でも 1.5 周分の点群データが得られていることを期待しており, 通常は一周分だから $\theta_E - \theta_S = 2\pi$ が成立する (半周分の増減を許容している)。変数 `halfPassed` は, 最初の角度 θ_S から半周 (180°) 以上回転したかどうかを示すフラグである。ROS の座標軸 $\tilde{Y}, \tilde{Z}, \tilde{X}$ は, LOAM の座標軸 X, Y, Z と対応しており, 実装でもそれが考慮されている。座標が NaN や無限大であるような点, LiDAR センサから距離が近すぎる点, 仰角が LiDAR センサの仕様から外れる点 (仰角が大きすぎ, あるいは小さすぎて, 対応するリングが存在しないような点) は除去される。

現在の i 番目の点に対し, `startOri` や `endOri` と同様の方法で方向 `ori` を計算する (これを θ_i とおく)。最初の角度 θ_S からまだ半周回転していない, 従って前半のデータを処理しているとき (`halfPassed` が `false`), $-\pi/2 < \theta_i - \theta_S < 3\pi/2$ となるように θ_i を調節する。 $\theta_i - \theta_S$ は, 現在の点が最初の角度からどの程度回転

しているかを表すため、負になる状況は考えづらい。但し θ_i に関しては何の仮定も置かない (任意の角度を取りうる) ため、 $\theta_i - \theta_S$ も $-\pi$ から $3\pi/2$ までの 2π の範囲で考えている。最初の角度から半周回転していれば ($\theta_i - \theta_S > \pi$ ならば), `halfPassed` を true にする。

最初の角度 θ_S から半周以上回転している、従って後半のデータを処理しているとき (`halfPassed` が true), $-3\pi/2 < \theta_i - \theta_E < \pi/2$ となるように θ_i を調節する。 $\theta_i - \theta_E$ は、現在の点が最後の角度からどの程度回転しているかを表すため、正になる状況は考えづらい (最後の角度よりも更に回転していれば正になる) が、やはり θ_i に関しては何の仮定も置かないため、実装内ではそうした状況も考慮している。

`relTime` は、最初の点の取得からの経過時間を表すオフセットである (これを r とおく)。最初の点が 0 で、最初の点から丁度一周したところで 1.0 となる。後述の通り、`relTime` は -0.5 から 1.5 の範囲を取りうる。 r は、始点と終点の方向 θ_S, θ_E , それから現在の方向 θ_i を使って次のように計算される。

$$r = \frac{\theta_i - \theta_S}{\theta_E - \theta_S} = \frac{\theta_i - \theta_E + (\theta_E - \theta_S)}{\theta_E - \theta_S} = \frac{\theta_i - \theta_E}{\theta_E - \theta_S} + \frac{\theta_E - \theta_S}{\theta_E - \theta_S} = 1 + \frac{\theta_i - \theta_E}{\theta_E - \theta_S} \quad (12)$$

前半のデータを処理しているとき、 $-\pi/2 < \theta_i - \theta_S < 3\pi/2$ かつ $\pi \leq \theta_E - \theta_S \leq 3\pi$ が成り立つ。通常の場合 ($\theta_E - \theta_S = 2\pi$) を考えると、 $\theta_i = \theta_S - \pi/2$ であれば $r = -0.25$, $\theta_i = \theta_S + 3\pi/2$ であれば $r = 0.75$, $\theta_i = \theta_S + \pi$ であれば $r = 0.5$, $\theta_i = \theta_S$ であれば $r = 0$ となる。データ取得に要した時間を t として、これを次のように言い換えられる。LiDAR が 360° 回転するのに t だけ掛かっているから、最初の点よりも $90^\circ (-\pi/2)$ だけ前側にある点、即ち最初の点よりも $0.25t$ だけ早く取得された点では、 $r = -0.25$ となる。同様に、最初の点よりも $270^\circ (3\pi/2)$ だけ後側にある点は、最初の点よりも $0.75t$ だけ遅く取得されているから $r = 0.75$ となる。最初の点から丁度半周した方向 (π) にある点の場合、最初の点よりも $0.5t$ だけ遅い時刻に取得されているから $r = 0.5$ である。このように r を使えば、各点が取得された時刻を正確に把握でき、点群データの歪みを補正できる。

点群が半周 (180°) 分のデータをもつとする ($\theta_E - \theta_S = \pi$)。最初の点よりも 90° だけ前側、最初の点、最初の点よりも 180° だけ後側、 270° だけ後側にある点、即ち $\theta_i - \theta_S = -\pi/2, 0, \pi, 3\pi/2$ である点は、最初の点よりも $-0.5t, 0, t, 1.5t$ だけずれて取得されているので、 $r = -0.5, 0, 1.0, 1.5$ となる。同様に、点群が 1.5 周 (540°) 分のデータをもつとする ($\theta_E - \theta_S = 3\pi$)。このとき、最初の点よりも 90° だけ前側、最初の点、最初の点よりも 180° だけ後側、 270° だけ後側にある点、即ち $\theta_i - \theta_S = -\pi/2, 0, \pi, 3\pi/2$ である点は、最初の点よりも $-t/6, 0, t/3, t/2$ だけずれて取得されているので、 $r = -1/6, 0, 1/3, 0.5$ となる。

上記は前半の点群に対する r の計算であったが、続いて後半の点群に対する処理を示す。後半のデータを処理しているとき、 $-3\pi/2 < \theta_i - \theta_E < \pi/2$ かつ $\pi \leq \theta_E - \theta_S \leq 3\pi$ となる。点群が一周 (360°) 分のデータをもつとする ($\theta_E - \theta_S = 2\pi$)。最後の点よりも 270° だけ前側、 180° だけ前側、最後の点、最後の点よりも 90° だけ後側にある点、即ち $\theta_i - \theta_E = -3\pi/2, -\pi, 0, \pi/2$ である点は、最後の点よりも $-0.75t, -0.5t, 0, 0.25t$ だけずれて取得されているので、 $r = 1 - 0.75 = 0.25, 1 - 0.5 = 0.5, 1 + 0 = 1, 1 + 0.25 = 1.25$ となって、正しい値が得られる (上式の最右項を利用する)。点群が半周 (180°) 分、1.5 周 (540°) 分のデータをもつ場合も同様に確認できる。上記の説明では r は割合であったが、実際には点群の取得間隔 `scanPeriod` が乗算されて、最初の点からの相対時間を表す。

Velodyne 社製の VLP-16 では仰角が 16 段階に分かれており、リングは同一の仰角をもった点のセットであるから、1 つの点群 `laserCloudIn` には計 16 個のリングが含まれる。各点が属するリングの番号を `MultiScanMapper` の `getRingForAngle()` により計算し、`scanID` に格納する。この `scanID` (リングの番号) と `relTime` (相対時間) を組み合わせることで、各点に固有の ID が得られ、これを `intensity` メンバに格納す

る (`intensity` は反射強度を示すので、このメンバの正しい使い方ではない)。固有の ID が得られる理由は、`scanID` は点の仰角、`relTime` は点の方位角に応じて定められ、同じ仰角と方位角をもつ点は、各点群内に 1 つしかないからである (図 1 参照)。 `intensity` メンバの整数部分が `scanID`、小数部分が `relTime` となるので、四捨五入によって `scanID`、そして `intensity` から `scanID` を引くことで `relTime` が復元できる。

各点の座標は、`BasicScanRegistration` の `projectPointToStartOfSweep()` を使って、点群 `laserCloudIn` のスキャン開始時刻 (スイープの開始時刻) `_scanTime` に揃えられる。このとき、スキャン開始時刻 `_scanTime` からの経過時間 `relTime` が必要である。`_scanTime`、`_scanTime + relTime` における IMU データが `_imuStart`、`_imuCur` であり、`_imuStart` から `_imuCur` への変化量を使って座標を補正する。スキャン開始時刻に揃えられた点群 `_laserCloudScans` は、論文の \mathcal{P}_{k+1} に対応する。

`_laserCloudScans` は各リングごとに分かれているが、`BasicScanRegistration` の `processScanlines()` で結合され、1 つの点群 `_laserCloud` となる (図 6 のように、各リングの始点と終点を指すインデックスが `_scanIndices` に格納される)。`processScanlines()` は点群 `_laserCloud` を 4 種類に分類し、`_cornerPointsSharp` (論文の \mathcal{E}_{k+1})、`_cornerPointsLessSharp`、`_surfacePointsFlat` (論文の \mathcal{H}_{k+1})、`_surfacePointsLessFlat` に格納する。`ScanRegistration` の `publishResult()` を呼び出すことで、前処理済みの点群 (上記 4 つの点群と `_laserCloud`) が Publish される。

5 BasicLaserOdometry クラス

5.1 BasicLaserOdometry::transformToStart()

与えられた点の座標 `pi` を、点群のスキャン開始時刻 (スイープの開始時刻) における座標に変換する。言い換えると、点の取得時点における LiDAR 座標系での座標 `pi` を、スキャン開始時刻における LiDAR 座標系での座標 `po` に変換する。この関数は `BasicScanRegistration` の `transformToStartIMU()` と同様の働きをもつ。`MultiScanRegistration` の `process()` で説明した通り、`ScanRegistration` ノードでの前処理によって、点群内の各点には個別の ID が割り当てられ、`intensity` フィールドに ID が格納されている。ID の整数部分はリングの番号 (これを基に仰角を復元できる)、小数部分は、点群のスキャン開始から、その点が実際に取得されるまでの時間差を表しているから、ID から整数部分を減算することで時間差を得られる (先述の `relTime` であり、`pi.intensity - int(pi.intensity)` と対応)。時間差を点群の取得間隔 `_scanPeriod` で割ることで、0 から 1 までの間に正規化する (変数 `s`)。従って、点群内の最初の点であれば (スキャン開始と同時に取得されたので) 0 となり、最後の点であれば (次の点群のスキャン開始直前、即ち現在の点群のスキャン開始から `_scanPeriod` だけ経過後に取得されたので) 1 となる。

`LaserOdometry` ノードでは、最新 ($k+1$ 番目) の点群のスキャン開始時刻 t_{k+1} に揃えられた直前 (k 番目) の点群 $\overline{\mathcal{P}}_k$ と、同じく t_{k+1} に揃えられた最新の点群 \mathcal{P}_{k+1} とのレジストレーション (重ね合わせ) によって、時刻 t_{k+1} から次の点群のスキャン開始時刻 t_{k+2} にかけての LiDAR の姿勢変化 T_{k+1}^L を求めようとする。3 次元空間での姿勢は 6 つのパラメータ $T_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$ で記述できる。 t_x, t_y, t_z は姿勢の並進成分、 $\theta_x, \theta_y, \theta_z$ は回転成分 (それぞれロール角、ピッチ角、ヨー角) である。LOAM の実装では以下の式で、時刻 t_{k+1} の LiDAR 座標系で表される点の座標 p を、時刻 t_{k+2} の LiDAR 座標系 p' に変換する (論文の式 (5) と同様だが微妙に異なる)。姿勢 T_{k+1}^L の回転成分 $-\theta_x, -\theta_y, -\theta_z$ は、時刻 t_{k+1} の LiDAR 座標系からみた、時刻 t_{k+2} の座標系の回転である (時刻 t_{k+2} の座標系が、時刻 t_{k+1} の座標系からどのくらい回転しているかを表す) から、点の座標を時刻 t_{k+1} から t_{k+2} の座標系へと回転するためには、回転行列の転置 $R(-\theta_x, -\theta_y, -\theta_z)^\top$ を

適用すればよい。LOAM の座標軸では $Z \rightarrow X \rightarrow Y$ の順で回転させるので、回転行列 $\mathbf{R}(-\theta_x, -\theta_y, -\theta_z)$ は $\mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z)$ と書ける。姿勢 \mathbf{T}_{k+1}^L の並進成分 $-t_x, -t_y, -t_z$ も、時刻 t_{k+1} の LiDAR 座標系からみた、時刻 t_{k+2} の LiDAR 座標系の位置であるから、点の座標を時刻 t_{k+1} から t_{k+2} の座標系へと変換するには、 $[-t_x, -t_y, -t_z]^\top$ を引けばよい。

$$\begin{aligned} \mathbf{p}' &= \mathbf{R}(-\theta_x, -\theta_y, -\theta_z)^\top \mathbf{p} - [-t_x, -t_y, -t_z]^\top \\ &= (\mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z))^\top \mathbf{p} + [t_x, t_y, t_z]^\top \end{aligned} \quad (13)$$

$$= (\mathbf{R}_z(\theta_z)\mathbf{R}_x(\theta_x)\mathbf{R}_y(\theta_y)) \mathbf{p} + [t_x, t_y, t_z]^\top \quad (14)$$

LOAM では、1 つの点群を取得する間、つまり 1 回のスイープの間は定速移動 (一定の速度と角速度) だと仮定される。時刻 t (但し $t_{k+1} \leq t \leq t_{k+2}$) の LiDAR 座標系における点の座標を $\mathbf{p}(t)$ とすると、定速移動であるから次の関係が成り立つ。

$$\begin{aligned} \mathbf{p}(t) &= \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z)^\top \mathbf{p}' + [st_x, st_y, st_z]^\top \\ &= (\mathbf{R}_y(-s\theta_y)\mathbf{R}_x(-s\theta_x)\mathbf{R}_z(-s\theta_z))^\top \mathbf{p}' + [st_x, st_y, st_z]^\top, \quad s = \frac{t - t_{k+1}}{t_{k+2} - t_{k+1}} \end{aligned} \quad (15)$$

$t = t_{k+2}$ ならば $s = 1$ であるから $\mathbf{p}(t) = \mathbf{p}'$ となり (時刻 t_{k+2} における LiDAR 座標系)、逆に $t = t_{k+1}$ であれば $s = 0$ ゆえ $\mathbf{p}(t) = \mathbf{p}$ が成り立つ (時刻 t_{k+1} における LiDAR 座標系)。

与えられた点 \mathbf{p}_i が最新 ($k+1$ 番目) の点群に属すると考えると、点 \mathbf{p}_i はスキャン開始時刻 t_{k+1} から $s * \text{_scanPeriod}$ だけ経過した時点で取得されている。ここで _scanPeriod は、 t_{k+1} から次の点群のスキャン開始時刻 t_{k+2} までの時間差 $t_{k+2} - t_{k+1}$ であるほか、点の取得時刻を t とすれば $s * \text{_scanPeriod}$ は $t - t_{k+1}$ に等しい。 s は 0 から 1 までの間で変化するから、 t は t_{k+1} から t_{k+2} までの範囲に収まる。上式における s は変数 s に、 $\mathbf{T}_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$ は _transform にそれぞれ対応する。また点の座標 \mathbf{p}_i は時刻 t の LiDAR 座標系で表されるため、上式の $\mathbf{p}(t)$ に対応する。 $\text{transformToStart}()$ で行いたいのは、時刻 t の LiDAR 座標系で表される点の座標 $\mathbf{p}(t)$ を、上式を使って時刻 t_{k+1} の LiDAR 座標系に戻して、その結果 \mathbf{p} を変数 \mathbf{p}_0 に格納することである。上式を \mathbf{p} について解けば以下ようになる。

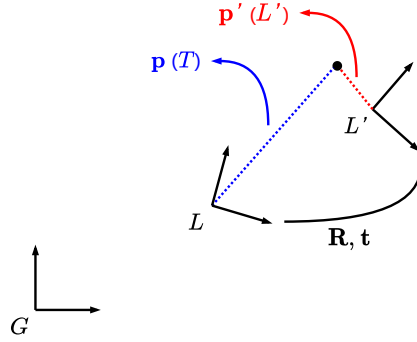
$$\mathbf{p} = (\mathbf{R}_y(-s\theta_y)\mathbf{R}_x(-s\theta_x)\mathbf{R}_z(-s\theta_z)) \left(\mathbf{p}(t) - [st_x, st_y, st_z]^\top \right) \quad (16)$$

$\text{transformToStart}()$ では最初に s を計算し、上式に倣って計算を行うことで、時刻 t_{k+1} の LiDAR 座標系における座標 \mathbf{p}_0 を求めている。

5.2 BasicLaserOdometry::transformToEnd()

与えられた点群 cloud に含まれる各点の座標を、点群のスキャン終了時刻 (スイープの終了時刻) における座標に変換する。言い換えると、各点の取得時点における LiDAR 座標系での座標を、スキャン終了時刻における LiDAR 座標系での座標に変換する (時刻を揃える)。 $\text{transformToEnd}()$ の前半部分は $\text{transformToStart}()$ と同一の処理である。 cloud が最新 ($k+1$ 番目) の点群だとすると、(16) 式を使った前半部分の処理によって、各点の座標 $\mathbf{p}(t)$ はスキャン開始時刻 t_{k+1} の LiDAR 座標系での座標 $\mathbf{p} = \mathbf{p}(t_{k+1})$ に変換される ($\text{transformToStart}()$ の説明を参照、図 9 の座標系 L)。更に (14) 式を使えば、スキャン終了時刻 t_{k+2} の LiDAR 座標系での座標 $\mathbf{p}' = \mathbf{p}(t_{k+2})$ が得られる (図 9 の座標系 L')。

MultiScanRegistration の $\text{process}()$ 内で点群の前処理を行うが、この際 BasicScanRegistration の $\text{projectPointToStartOfSweep}()$ を通じて $\text{transformToStartIMU}()$ を呼び出す。このメソッド内で



G : グローバル座標系
 L : t_{k+1} におけるLiDAR座標系
 L' : t_{k+2} におけるLiDAR座標系
 $\mathbf{R} = \mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z)$
 $\mathbf{t} = [-t_x, -t_y, -t_z]^\top$
 $\mathbf{T}_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$

図7 時刻 t_{k+1}, t_{k+2} の LiDAR 座標系における点の座標 p, p'

は、各点の座標に `_imuPositionShift` を加えることで、点の取得時刻 t における LiDAR 座標系から、時刻 t_{k+1} から t までの間定速で移動していた場合の (時刻 t における) 座標系へと補正している。 p' は前処理済みの点群内の点であるから、時刻 t_{k+1} から t_{k+2} までの間定速移動していた場合の (時刻 t_{k+2} における) LiDAR 座標系として解釈される (図8の座標系 S' , 図9の座標系 L')。従って p' から `_imuShiftFromStart(BasicScanRegistration の updateIMUTransform() 内で計算)` を引くことで、(補正する前の) 時刻 t_{k+2} における LiDAR 座標系へと座標を戻せる (図8の座標系 E , 図9の座標系 L'')。ここで `_imuShiftFromStart` は、座標系 $S'(L')$ からみた $E(L'')$ の位置であるから、点の座標 p' を座標系 $S'(L')$ から $E(L'')$ へと移すためには `_imuShiftFromStart` を引けばよい。

またグローバル座標系 G から S' への座標系の回転はオイラー角 $\tilde{\phi}_s, \tilde{\theta}_s, \tilde{\psi}_s$ (`_imuRollStart, _imuPitchStart, _imuYawStart`), G から E への座標系の回転はオイラー角 $\tilde{\phi}_e, \tilde{\theta}_e, \tilde{\psi}_e$ (`_imuRollEnd, _imuPitchEnd, _imuYawEnd`) から得られる。これらは `BasicScanRegistration の updateIMUTransform() 内で計算` される `_imuStart` と `_imuCur` に対応しており、ROS の座標軸 $\tilde{X}, \tilde{Y}, \tilde{Z}$ 周りの回転を表す (ROS では $\tilde{X} \rightarrow \tilde{Y} \rightarrow \tilde{Z}$ の順に回転させる)。点の座標を座標系 S' から $E(L'$ から L'') へと移すような回転行列は、点の座標を S' から G , G から E へ移すような回転行列を合成すれば得られる。回転行列 $\tilde{\mathbf{R}}_z(\tilde{\psi}_s)\tilde{\mathbf{R}}_y(\tilde{\theta}_s)\tilde{\mathbf{R}}_x(\tilde{\phi}_s) = \mathbf{R}_y(\tilde{\psi}_s)\mathbf{R}_x(\tilde{\theta}_s)\mathbf{R}_z(\tilde{\phi}_s)$ と $\tilde{\mathbf{R}}_z(\tilde{\psi}_e)\tilde{\mathbf{R}}_y(\tilde{\theta}_e)\tilde{\mathbf{R}}_x(\tilde{\phi}_e) = \mathbf{R}_y(\tilde{\psi}_e)\mathbf{R}_x(\tilde{\theta}_e)\mathbf{R}_z(\tilde{\phi}_e)$ はそれぞれ、座標系を G から S' , G から E へと回転させるので、点の座標を S' から G , E から G へと変換する。 E から G へと変換する回転行列を転置すれば、 G から E へと変換する回転行列が得られる。従って、求める回転行列は次のようになり (S' から E , L' から L'')、 p' から `_imuShiftFromStart` に引いたものに適用すればよい。

$$\begin{aligned}
& \left(\mathbf{R}_y(\tilde{\psi}_e)\mathbf{R}_x(\tilde{\theta}_e)\mathbf{R}_z(\tilde{\phi}_e) \right)^\top \mathbf{R}_y(\tilde{\psi}_s)\mathbf{R}_x(\tilde{\theta}_s)\mathbf{R}_z(\tilde{\phi}_s) \\
& = \left(\mathbf{R}_z(\tilde{\phi}_e)^\top \mathbf{R}_x(\tilde{\theta}_e)^\top \mathbf{R}_y(\tilde{\psi}_e)^\top \right) \mathbf{R}_y(\tilde{\psi}_s)\mathbf{R}_x(\tilde{\theta}_s)\mathbf{R}_z(\tilde{\phi}_s)
\end{aligned}$$

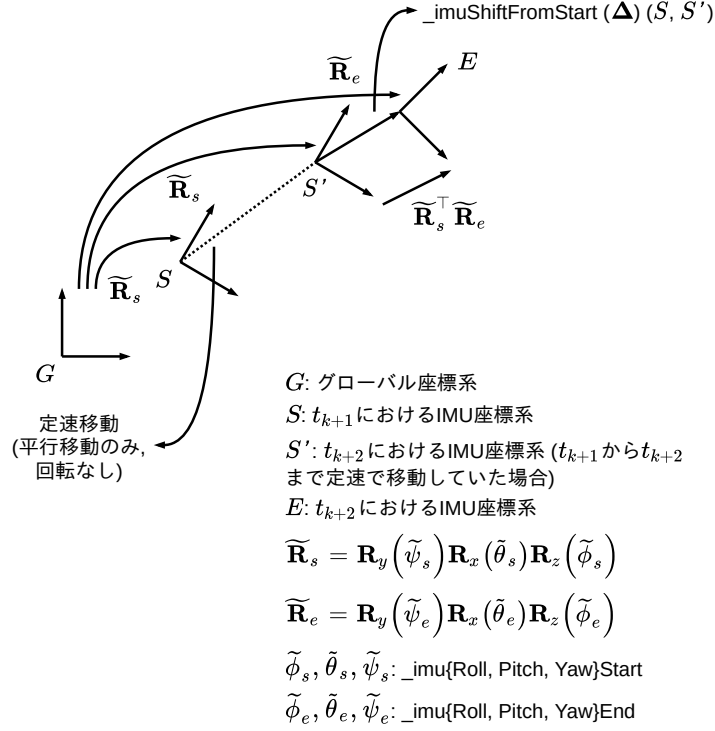


図8 時刻 t_{k+1}, t_{k+2} における IMU 座標系

$$= \left(\mathbf{R}_z(-\widetilde{\phi}_e)\mathbf{R}_x(-\widetilde{\theta}_e)\mathbf{R}_y(-\widetilde{\psi}_e) \right) \mathbf{R}_y(\widetilde{\psi}_s)\mathbf{R}_x(\widetilde{\theta}_s)\mathbf{R}_z(\widetilde{\phi}_s)$$

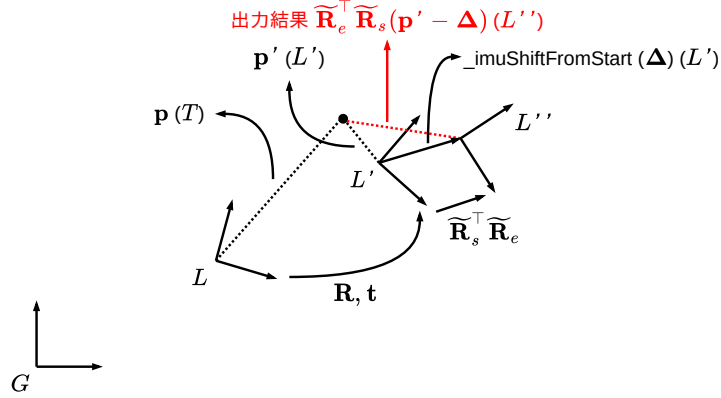
上記の回転操作は `rotateZXY()` と `rotateYXZ()` の呼び出しによって実現される。

このようにして、点群 `cloud` に含まれる各点の座標を、点群のスキャン終了時刻 (スイープの終了時刻) に揃えたうえで、`BasicScanRegistration` の `transformToStartIMU()` で行われた、定速移動の仮定に基づく座標の補正を取り除いて、本来の点の座標に戻している。 `transformToEnd()` は、`ScanRegistration` ノードから受け取った最新 ($k+1$ 番目) の点群 `_cornerPointsLessSharp` (それほど顕著ではないコーナー)、`_surfPointsLessFlat` (それほど顕著ではない平面)、`_laserCloud` (点群全体) に対して適用され、その後 `LaserOdometry` の `publishResult()` を通じて `LaserMapping` ノードに渡される。 `LaserMapping` では、スキャン終了時刻 t_{k+2} に揃えられ、かつ座標の補正を行っていない最新の点群 $\overline{\mathcal{P}}_{k+1}$ を受け取って、マッピングを実行する。

5.3 BasicLaserOdometry::pluginIMURotation()

オイラー角 $\phi_b^c, \theta_b^c, \psi_b^c$ (`bcx, bcy, bcz`), $\phi_b^l, \theta_b^l, \psi_b^l$ (`blx, bly, blz`), $\phi_a^l, \theta_a^l, \psi_a^l$ (`alx, aly, alz`) から作成した回転行列 $\mathbf{R}_b^c = \mathbf{R}_y(\theta_b^c)\mathbf{R}_x(\phi_b^c)\mathbf{R}_z(\psi_b^c)$, $\mathbf{R}_b^l = \mathbf{R}_y(\theta_b^l)\mathbf{R}_x(\phi_b^l)\mathbf{R}_z(\psi_b^l)$, $\mathbf{R}_a^l = \mathbf{R}_y(\theta_a^l)\mathbf{R}_x(\phi_a^l)\mathbf{R}_z(\psi_a^l)$ があるとす。 これらを合成した回転行列 $\mathbf{R}_a^c = \mathbf{R}_b^c (\mathbf{R}_b^l)^\top \mathbf{R}_a^l$ に対応するオイラー角 $\phi_a^c, \theta_a^c, \psi_a^c$ (`acx, acy, acz`) を計算する。 オイラー角は LOAM の座標軸で $Z \rightarrow X \rightarrow Y$ の順に回転させる。

下準備として、オイラー角 ϕ, θ, ψ に対応する回転行列 $\mathbf{R} = \mathbf{R}_y(\theta)\mathbf{R}_x(\phi)\mathbf{R}_z(\psi)$ を要素ごとに書き下してみ



- G : グローバル座標系
 L : t_{k+1} におけるLiDAR座標系
 L' : t_{k+2} におけるLiDAR座標系 (補正後)
 L'' : t_{k+2} におけるLiDAR座標系 (補正前)
 $\mathbf{R} = \mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z)$
 $\mathbf{t} = [-t_x, -t_y, -t_z]^\top$
 $\mathbf{T}_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$ (_transform)

図9 時刻 t_{k+1}, t_{k+2} における LiDAR 座標系

よう.

$$\begin{aligned}
 \mathbf{R} &= \mathbf{R}_y(\theta)\mathbf{R}_x(\phi)\mathbf{R}_z(\psi) \\
 &= \mathbf{R}_y(\theta) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \mathbf{R}_y(\theta) \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \cos \phi \sin \psi & \cos \phi \cos \psi & -\sin \phi \\ \sin \phi \sin \psi & \sin \phi \cos \psi & \cos \phi \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \cos \phi \sin \psi & \cos \phi \cos \psi & -\sin \phi \\ \sin \phi \sin \psi & \sin \phi \cos \psi & \cos \phi \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta \cos \psi + \sin \theta \sin \phi \sin \psi & -\cos \theta \sin \psi + \sin \theta \sin \phi \cos \psi & \sin \theta \cos \phi \\ \cos \phi \sin \psi & \cos \phi \cos \psi & -\sin \phi \\ -\sin \theta \cos \psi + \cos \theta \sin \phi \sin \psi & \sin \theta \sin \psi + \cos \theta \sin \phi \cos \psi & \cos \theta \cos \phi \end{bmatrix} \quad (17)
 \end{aligned}$$

回転行列の転置 \mathbf{R}^\top は次のようになる.

$$\mathbf{R}^\top = \begin{bmatrix} \cos \theta \cos \psi + \sin \theta \sin \phi \sin \psi & \cos \phi \sin \psi & -\sin \theta \cos \psi + \cos \theta \sin \phi \sin \psi \\ -\cos \theta \sin \psi + \sin \theta \sin \phi \cos \psi & \cos \phi \cos \psi & \sin \theta \sin \psi + \cos \theta \sin \phi \cos \psi \\ \sin \theta \cos \phi & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (18)$$

回転行列 \mathbf{R} からオイラー角 ϕ, θ, ψ を次のように復元できる (R_{ij} は, 行列 \mathbf{R} の i 行 j 列目の要素を指す).

$$\phi = -\arcsin R_{12}, \quad \theta = \arctan\left(\frac{R_{02}}{R_{22}}\right), \quad \psi = \arctan\left(\frac{R_{10}}{R_{11}}\right) \quad (19)$$

これらを使って、回転行列 $R_b^c, (R_b^l)^\top, R_a^l$ は次のように書ける。

$$\begin{aligned}
R_b^c &= \begin{bmatrix} \cos \theta_b^c \cos \psi_b^c + \sin \theta_b^c \sin \phi_b^c \sin \psi_b^c & -\cos \theta_b^c \sin \psi_b^c + \sin \theta_b^c \sin \phi_b^c \cos \psi_b^c & \sin \theta_b^c \cos \phi_b^c \\ \cos \phi_b^c \sin \psi_b^c & \cos \phi_b^c \cos \psi_b^c & -\sin \phi_b^c \\ -\sin \theta_b^c \cos \psi_b^c + \cos \theta_b^c \sin \phi_b^c \sin \psi_b^c & \sin \theta_b^c \sin \psi_b^c + \cos \theta_b^c \sin \phi_b^c \cos \psi_b^c & \cos \theta_b^c \cos \phi_b^c \end{bmatrix} \\
(R_b^l)^\top &= \begin{bmatrix} \cos \theta_b^l \cos \psi_b^l + \sin \theta_b^l \sin \phi_b^l \sin \psi_b^l & \cos \phi_b^l \sin \psi_b^l & -\sin \theta_b^l \cos \psi_b^l + \cos \theta_b^l \sin \phi_b^l \sin \psi_b^l \\ -\cos \theta_b^l \sin \psi_b^l + \sin \theta_b^l \sin \phi_b^l \cos \psi_b^l & \cos \phi_b^l \cos \psi_b^l & \sin \theta_b^l \sin \psi_b^l + \cos \theta_b^l \sin \phi_b^l \cos \psi_b^l \\ \sin \theta_b^l \cos \phi_b^l & -\sin \phi_b^l & \cos \theta_b^l \cos \phi_b^l \end{bmatrix} \\
R_a^l &= \begin{bmatrix} \cos \theta_a^l \cos \psi_a^l + \sin \theta_a^l \sin \phi_a^l \sin \psi_a^l & -\cos \theta_a^l \sin \psi_a^l + \sin \theta_a^l \sin \phi_a^l \cos \psi_a^l & \sin \theta_a^l \cos \phi_a^l \\ \cos \phi_a^l \sin \psi_a^l & \cos \phi_a^l \cos \psi_a^l & -\sin \phi_a^l \\ -\sin \theta_a^l \cos \psi_a^l + \cos \theta_a^l \sin \phi_a^l \sin \psi_a^l & \sin \theta_a^l \sin \psi_a^l + \cos \theta_a^l \sin \phi_a^l \cos \psi_a^l & \cos \theta_a^l \cos \phi_a^l \end{bmatrix}
\end{aligned}$$

行列 $R_a^c = R_b^c (R_b^l)^\top R_a^l$ からオイラー角 $\phi_a^c, \theta_a^c, \psi_a^c$ を復元するためには、行列 R_a^c のうち 5 つの要素 $R_{a12}^c, R_{a02}^c, R_{a22}^c, R_{a10}^c, R_{a11}^c$ を求めればよい。ロール角 ϕ_a^c を復元するため、最初に R_{a12}^c を計算してみよう。

$$\begin{aligned}
R_{a12}^c &= R_{b10}^c \left((R_b^l)_{00}^\top R_{a02}^l + (R_b^l)_{01}^\top R_{a12}^l + (R_b^l)_{02}^\top R_{a22}^l \right) \\
&\quad + R_{b11}^c \left((R_b^l)_{10}^\top R_{a02}^l + (R_b^l)_{11}^\top R_{a12}^l + (R_b^l)_{12}^\top R_{a22}^l \right) \\
&\quad + R_{b12}^c \left((R_b^l)_{20}^\top R_{a02}^l + (R_b^l)_{21}^\top R_{a12}^l + (R_b^l)_{22}^\top R_{a22}^l \right) \\
&= \cos \phi_b^c \sin \psi_b^c \left((\cos \theta_b^l \cos \psi_b^l + \sin \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\sin \theta_a^l \cos \phi_a^l) \right. \\
&\quad \left. - (\cos \phi_b^l \sin \psi_b^l) (\sin \phi_a^l) + (-\sin \theta_b^l \cos \psi_b^l + \cos \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\cos \theta_a^l \cos \phi_a^l) \right) \\
&\quad + \cos \phi_b^c \cos \psi_b^c \left((-\cos \theta_b^l \sin \psi_b^l + \sin \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\sin \theta_a^l \cos \phi_a^l) \right. \\
&\quad \left. - (\cos \phi_b^l \cos \psi_b^l) (\sin \phi_a^l) + (\sin \theta_b^l \sin \psi_b^l + \cos \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\cos \theta_a^l \cos \phi_a^l) \right) \\
&\quad - \sin \phi_b^c \left((\sin \theta_b^l \cos \phi_b^l) (\sin \theta_a^l \cos \phi_a^l) + (\sin \phi_b^l) (\sin \phi_a^l) + (\cos \theta_b^l \cos \phi_b^l) (\cos \theta_a^l \cos \phi_a^l) \right)
\end{aligned}$$

続いて、ピッチ角 θ_a^c を復元するため、 R_{a02}^c, R_{a22}^c を計算してみよう。

$$\begin{aligned}
R_{a02}^c &= R_{b00}^c \left((R_b^l)_{00}^\top R_{a02}^l + (R_b^l)_{01}^\top R_{a12}^l + (R_b^l)_{02}^\top R_{a22}^l \right) \\
&\quad + R_{b01}^c \left((R_b^l)_{10}^\top R_{a02}^l + (R_b^l)_{11}^\top R_{a12}^l + (R_b^l)_{12}^\top R_{a22}^l \right) \\
&\quad + R_{b02}^c \left((R_b^l)_{20}^\top R_{a02}^l + (R_b^l)_{21}^\top R_{a12}^l + (R_b^l)_{22}^\top R_{a22}^l \right) \\
&= (\cos \theta_b^c \cos \psi_b^c + \sin \theta_b^c \sin \phi_b^c \sin \psi_b^c) \left((\cos \theta_b^l \cos \psi_b^l + \sin \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\sin \theta_a^l \cos \phi_a^l) \right. \\
&\quad \left. - (\cos \phi_b^l \sin \psi_b^l) (\sin \phi_a^l) + (-\sin \theta_b^l \cos \psi_b^l + \cos \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\cos \theta_a^l \cos \phi_a^l) \right) \\
&\quad + (-\cos \theta_b^c \sin \psi_b^c + \sin \theta_b^c \sin \phi_b^c \cos \psi_b^c) \left((-\cos \theta_b^l \sin \psi_b^l + \sin \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\sin \theta_a^l \cos \phi_a^l) \right. \\
&\quad \left. - (\cos \phi_b^l \cos \psi_b^l) (\sin \phi_a^l) + (\sin \theta_b^l \sin \psi_b^l + \cos \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\cos \theta_a^l \cos \phi_a^l) \right) \\
&\quad + (\sin \theta_b^c \cos \phi_b^c) \left((\sin \theta_b^l \cos \phi_b^l) (\sin \theta_a^l \cos \phi_a^l) + (\sin \phi_b^l) (\sin \phi_a^l) + (\cos \theta_b^l \cos \phi_b^l) (\cos \theta_a^l \cos \phi_a^l) \right)
\end{aligned}$$

$$\begin{aligned}
R_{a22}^c &= R_{b20}^c \left((R_b^l)_{00}^\top R_{a02}^l + (R_b^l)_{01}^\top R_{a12}^l + (R_b^l)_{02}^\top R_{a22}^l \right) \\
&\quad + R_{b21}^c \left((R_b^l)_{10}^\top R_{a02}^l + (R_b^l)_{11}^\top R_{a12}^l + (R_b^l)_{12}^\top R_{a22}^l \right) \\
&\quad + R_{b22}^c \left((R_b^l)_{20}^\top R_{a02}^l + (R_b^l)_{21}^\top R_{a12}^l + (R_b^l)_{22}^\top R_{a22}^l \right) \\
&= (-\sin \theta_b^c \cos \psi_b^c + \cos \theta_b^c \sin \phi_b^c \sin \psi_b^c) \left((\cos \theta_b^l \cos \psi_b^l + \sin \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\sin \theta_a^l \cos \phi_a^l) \right. \\
&\quad \left. - (\cos \phi_b^l \sin \psi_b^l) (\sin \phi_a^l) + (-\sin \theta_b^l \cos \psi_b^l + \cos \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\cos \theta_a^l \cos \phi_a^l) \right)
\end{aligned}$$

$$\begin{aligned}
& + (\sin \theta_b^c \sin \psi_b^c + \cos \theta_b^c \sin \phi_b^c \cos \psi_b^c) ((-\cos \theta_b^l \sin \psi_b^l + \sin \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\sin \theta_a^l \cos \phi_a^l) \\
& \quad - (\cos \phi_b^l \cos \psi_b^l) (\sin \phi_a^l) + (\sin \theta_b^l \sin \psi_b^l + \cos \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\cos \theta_a^l \cos \phi_a^l)) \\
& + (\cos \theta_b^c \cos \phi_b^c) ((\sin \theta_b^l \cos \phi_b^l) (\sin \theta_a^l \cos \phi_a^l) + (\sin \phi_b^l) (\sin \phi_a^l) + (\cos \theta_b^l \cos \phi_b^l) (\cos \theta_a^l \cos \phi_a^l))
\end{aligned}$$

最後に、ヨ一角 ψ_a^c を復元するため、 $\mathbf{R}_{a10}^c, \mathbf{R}_{a11}^c$ を復元してみよう。

$$\begin{aligned}
\mathbf{R}_{a10}^c & = \mathbf{R}_{b10}^c \left((\mathbf{R}_b^l)_{00}^\top \mathbf{R}_{a00}^l + (\mathbf{R}_b^l)_{01}^\top \mathbf{R}_{a10}^l + (\mathbf{R}_b^l)_{02}^\top \mathbf{R}_{a20}^l \right) \\
& + \mathbf{R}_{b11}^c \left((\mathbf{R}_b^l)_{10}^\top \mathbf{R}_{a00}^l + (\mathbf{R}_b^l)_{11}^\top \mathbf{R}_{a10}^l + (\mathbf{R}_b^l)_{12}^\top \mathbf{R}_{a20}^l \right) \\
& + \mathbf{R}_{b12}^c \left((\mathbf{R}_b^l)_{20}^\top \mathbf{R}_{a00}^l + (\mathbf{R}_b^l)_{21}^\top \mathbf{R}_{a10}^l + (\mathbf{R}_b^l)_{22}^\top \mathbf{R}_{a20}^l \right) \\
& = \cos \phi_b^c \sin \psi_b^c ((\cos \theta_b^l \cos \psi_b^l + \sin \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\cos \theta_a^l \cos \psi_a^l + \sin \theta_a^l \sin \phi_a^l \sin \psi_a^l) \\
& \quad + (\cos \phi_b^l \sin \psi_b^l) (\cos \phi_a^l \sin \psi_a^l) \\
& \quad + (-\sin \theta_b^l \cos \psi_b^l + \cos \theta_b^l \sin \phi_b^l \sin \psi_b^l) (-\sin \theta_a^l \cos \psi_a^l + \cos \theta_a^l \sin \phi_a^l \sin \psi_a^l)) \\
& + \cos \phi_b^c \cos \psi_b^c ((-\cos \theta_b^l \sin \psi_b^l + \sin \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\cos \theta_a^l \cos \psi_a^l + \sin \theta_a^l \sin \phi_a^l \sin \psi_a^l) \\
& \quad + (\cos \phi_b^l \cos \psi_b^l) (\cos \phi_a^l \sin \psi_a^l) \\
& \quad + (\sin \theta_b^l \sin \psi_b^l + \cos \theta_b^l \sin \phi_b^l \cos \psi_b^l) (-\sin \theta_a^l \cos \psi_a^l + \cos \theta_a^l \sin \phi_a^l \sin \psi_a^l)) \\
& - \sin \phi_b^c ((\sin \theta_b^l \cos \phi_b^l) (\cos \theta_a^l \cos \psi_a^l + \sin \theta_a^l \sin \phi_a^l \sin \psi_a^l) \\
& \quad - (\sin \phi_b^l) (\cos \phi_a^l \sin \psi_a^l) + (\cos \theta_b^l \cos \phi_b^l) (-\sin \theta_a^l \cos \psi_a^l + \cos \theta_a^l \sin \phi_a^l \sin \psi_a^l))
\end{aligned}$$

$$\begin{aligned}
\mathbf{R}_{a11}^c & = \mathbf{R}_{b10}^c \left((\mathbf{R}_b^l)_{00}^\top \mathbf{R}_{a01}^l + (\mathbf{R}_b^l)_{01}^\top \mathbf{R}_{a11}^l + (\mathbf{R}_b^l)_{02}^\top \mathbf{R}_{a21}^l \right) \\
& + \mathbf{R}_{b11}^c \left((\mathbf{R}_b^l)_{10}^\top \mathbf{R}_{a01}^l + (\mathbf{R}_b^l)_{11}^\top \mathbf{R}_{a11}^l + (\mathbf{R}_b^l)_{12}^\top \mathbf{R}_{a21}^l \right) \\
& + \mathbf{R}_{b12}^c \left((\mathbf{R}_b^l)_{20}^\top \mathbf{R}_{a01}^l + (\mathbf{R}_b^l)_{21}^\top \mathbf{R}_{a11}^l + (\mathbf{R}_b^l)_{22}^\top \mathbf{R}_{a21}^l \right) \\
& = \cos \phi_b^c \sin \psi_b^c ((\cos \theta_b^l \cos \psi_b^l + \sin \theta_b^l \sin \phi_b^l \sin \psi_b^l) (-\cos \theta_a^l \sin \psi_a^l + \sin \theta_a^l \sin \phi_a^l \cos \psi_a^l) \\
& \quad + (\cos \phi_b^l \sin \psi_b^l) (\cos \phi_a^l \cos \psi_a^l) \\
& \quad + (-\sin \theta_b^l \cos \psi_b^l + \cos \theta_b^l \sin \phi_b^l \sin \psi_b^l) (\sin \theta_a^l \sin \psi_a^l + \cos \theta_a^l \sin \phi_a^l \cos \psi_a^l)) \\
& + \cos \phi_b^c \cos \psi_b^c ((-\cos \theta_b^l \sin \psi_b^l + \sin \theta_b^l \sin \phi_b^l \cos \psi_b^l) (-\cos \theta_a^l \sin \psi_a^l + \sin \theta_a^l \sin \phi_a^l \cos \psi_a^l) \\
& \quad + (\cos \phi_b^l \cos \psi_b^l) (\cos \phi_a^l \cos \psi_a^l) \\
& \quad + (\sin \theta_b^l \sin \psi_b^l + \cos \theta_b^l \sin \phi_b^l \cos \psi_b^l) (\sin \theta_a^l \sin \psi_a^l + \cos \theta_a^l \sin \phi_a^l \cos \psi_a^l)) \\
& - \sin \phi_b^c ((\sin \theta_b^l \cos \phi_b^l) (-\cos \theta_a^l \sin \psi_a^l + \sin \theta_a^l \sin \phi_a^l \cos \psi_a^l) \\
& \quad - (\sin \phi_b^l) (\cos \phi_a^l \cos \psi_a^l) + (\cos \theta_b^l \cos \phi_b^l) (\sin \theta_a^l \sin \psi_a^l + \cos \theta_a^l \sin \phi_a^l \cos \psi_a^l))
\end{aligned}$$

これらを基に、オイラー角 $\phi_a^c, \theta_a^c, \psi_a^c$ を次のように復元できる。

$$\phi_a^c = -\arcsin \mathbf{R}_{a12}^c, \quad \theta_a^c = \arctan \left(\frac{\mathbf{R}_{a02}^c}{\mathbf{R}_{a22}^c} \right), \quad \psi_a^c = \arctan \left(\frac{\mathbf{R}_{a10}^c}{\mathbf{R}_{a11}^c} \right)$$

ソースコード内の変数とは次のように対応している。

$$\begin{aligned}
\text{sbcx} & = \sin \phi_b^c, & \text{cbcx} & = \cos \phi_b^c, & \text{sbcy} & = \sin \theta_b^c, & \text{cbcy} & = \cos \theta_b^c, & \text{sbcz} & = \sin \psi_b^c, & \text{cbcz} & = \cos \psi_b^c \\
\text{sblx} & = \sin \phi_b^l, & \text{cblx} & = \cos \phi_b^l, & \text{sbyl} & = \sin \theta_b^l, & \text{cbyl} & = \cos \theta_b^l, & \text{sblz} & = \sin \psi_b^l, & \text{cblz} & = \cos \psi_b^l \\
\text{salx} & = \sin \phi_a^l, & \text{calx} & = \cos \phi_a^l, & \text{saly} & = \sin \theta_a^l, & \text{caly} & = \cos \theta_a^l, & \text{salz} & = \sin \psi_a^l, & \text{calz} & = \cos \psi_a^l
\end{aligned}$$

$$\begin{aligned}
\text{srx} &= \mathbf{R}_{a12}^c, & \text{acx} &= \phi_a^c \\
\text{srycrx} &= \mathbf{R}_{a02}^c, & \text{crycrx} &= \mathbf{R}_{a22}^c, & \text{acy} &= \theta_a^c \\
\text{srzcrx} &= \mathbf{R}_{a10}^c, & \text{crzcrx} &= \mathbf{R}_{a11}^c, & \text{acz} &= \psi_a^c
\end{aligned}$$

5.4 BasicLaserOdometry::accumulateRotation()

オイラー角 ϕ^c, θ^c, ψ^c (cx, cy, cz), ϕ^l, θ^l, ψ^l (lx, ly, lz) から作成した回転行列 $\mathbf{R}^c = \mathbf{R}_y(\theta^c)\mathbf{R}_x(\phi^c)\mathbf{R}_z(\psi^c)$, $\mathbf{R}^l = \mathbf{R}_y(\theta^l)\mathbf{R}_x(\phi^l)\mathbf{R}_z(\psi^l)$ があるとす。この2つを合成した回転行列 $\mathbf{R}^o = \mathbf{R}^c\mathbf{R}^l$ に対応するオイラー角 ϕ^o, θ^o, ψ^o (ox, oy, oz) を計算する。オイラー角は LOAM の座標軸で $Z \rightarrow X \rightarrow Y$ の順に回転させる。

回転行列 $\mathbf{R}^c, \mathbf{R}^l$ を要素ごとに書き下すと、次のようになる。

$$\mathbf{R}^c = \begin{bmatrix} \cos \theta^c \cos \psi^c + \sin \theta^c \sin \phi^c \sin \psi^c & -\cos \theta^c \sin \psi^c + \sin \theta^c \sin \phi^c \cos \psi^c & \sin \theta^c \cos \phi^c \\ \cos \phi^c \sin \psi^c & \cos \phi^c \cos \psi^c & -\sin \phi^c \\ -\sin \theta^c \cos \psi^c + \cos \theta^c \sin \phi^c \sin \psi^c & \sin \theta^c \sin \psi^c + \cos \theta^c \sin \phi^c \cos \psi^c & \cos \theta^c \cos \phi^c \end{bmatrix}$$

$$\mathbf{R}^l = \begin{bmatrix} \cos \theta^l \cos \psi^l + \sin \theta^l \sin \phi^l \sin \psi^l & -\cos \theta^l \sin \psi^l + \sin \theta^l \sin \phi^l \cos \psi^l & \sin \theta^l \cos \phi^l \\ \cos \phi^l \sin \psi^l & \cos \phi^l \cos \psi^l & -\sin \phi^l \\ -\sin \theta^l \cos \psi^l + \cos \theta^l \sin \phi^l \sin \psi^l & \sin \theta^l \sin \psi^l + \cos \theta^l \sin \phi^l \cos \psi^l & \cos \theta^l \cos \phi^l \end{bmatrix}$$

行列 $\mathbf{R}^o = \mathbf{R}^c\mathbf{R}^l$ からオイラー角 ϕ^o, θ^o, ψ^o を復元するためには、 \mathbf{R}^o のうち5つの要素 $\mathbf{R}_{12}^o, \mathbf{R}_{02}^o, \mathbf{R}_{22}^o, \mathbf{R}_{10}^o, \mathbf{R}_{11}^o$ を求めればよい。これら5つの要素は次のように計算できる。

$$\begin{aligned}
\mathbf{R}_{12}^o &= \mathbf{R}_{10}^c\mathbf{R}_{02}^l + \mathbf{R}_{11}^c\mathbf{R}_{12}^l + \mathbf{R}_{12}^c\mathbf{R}_{22}^l \\
&= (\cos \phi^c \sin \psi^c) (\sin \theta^l \cos \phi^l) - (\cos \phi^c \cos \psi^c) (\sin \phi^l) - (\sin \phi^c) (\cos \theta^l \cos \phi^l) \\
\mathbf{R}_{02}^o &= \mathbf{R}_{00}^c\mathbf{R}_{02}^l + \mathbf{R}_{01}^c\mathbf{R}_{12}^l + \mathbf{R}_{02}^c\mathbf{R}_{22}^l \\
&= (\cos \theta^c \cos \psi^c + \sin \theta^c \sin \phi^c \sin \psi^c) (\sin \theta^l \cos \phi^l) \\
&\quad - (-\cos \theta^c \sin \psi^c + \sin \theta^c \sin \phi^c \cos \psi^c) (\sin \phi^l) + (\sin \theta^c \cos \phi^c) (\cos \theta^l \cos \phi^l) \\
\mathbf{R}_{22}^o &= \mathbf{R}_{20}^c\mathbf{R}_{02}^l + \mathbf{R}_{21}^c\mathbf{R}_{12}^l + \mathbf{R}_{22}^c\mathbf{R}_{22}^l \\
&= (-\sin \theta^c \cos \psi^c + \cos \theta^c \sin \phi^c \sin \psi^c) (\sin \theta^l \cos \phi^l) \\
&\quad - (\sin \theta^c \sin \psi^c + \cos \theta^c \sin \phi^c \cos \psi^c) (\sin \phi^l) + (\cos \theta^c \cos \phi^c) (\cos \theta^l \cos \phi^l) \\
\mathbf{R}_{10}^o &= \mathbf{R}_{10}^c\mathbf{R}_{00}^l + \mathbf{R}_{11}^c\mathbf{R}_{10}^l + \mathbf{R}_{12}^c\mathbf{R}_{20}^l \\
&= (\cos \phi^c \sin \psi^c) (\cos \theta^l \cos \psi^l + \sin \theta^l \sin \phi^l \sin \psi^l) \\
&\quad + (\cos \phi^c \cos \psi^c) (\cos \phi^l \sin \psi^l) - (\sin \phi^c) (-\sin \theta^l \cos \psi^l + \cos \theta^l \sin \phi^l \sin \psi^l) \\
\mathbf{R}_{11}^o &= \mathbf{R}_{10}^c\mathbf{R}_{01}^l + \mathbf{R}_{11}^c\mathbf{R}_{11}^l + \mathbf{R}_{12}^c\mathbf{R}_{21}^l \\
&= (\cos \phi^c \sin \psi^c) (-\cos \theta^l \sin \psi^l + \sin \theta^l \sin \phi^l \cos \psi^l) \\
&\quad + (\cos \phi^c \cos \psi^c) (\cos \phi^l \cos \psi^l) - (\sin \phi^c) (\sin \theta^l \sin \psi^l + \cos \theta^l \sin \phi^l \cos \psi^l)
\end{aligned}$$

これらを基に、オイラー角 ϕ^o, θ^o, ψ^o を次のように復元できる。

$$\phi^o = -\arcsin \mathbf{R}_{12}^o, \quad \theta^o = \arctan \left(\frac{\mathbf{R}_{02}^o}{\mathbf{R}_{22}^o} \right), \quad \psi^o = \arctan \left(\frac{\mathbf{R}_{10}^o}{\mathbf{R}_{11}^o} \right)$$

ソースコード内の変数とは次のように対応している。

$$\begin{aligned}
\text{srx} &= \mathbf{R}_{12}^o, & \text{ox} &= \phi^o \\
\text{srycrx} &= \mathbf{R}_{02}^o, & \text{crycrx} &= \mathbf{R}_{22}^o, & \text{oy} &= \theta^o \\
\text{srzcrx} &= \mathbf{R}_{10}^o, & \text{crzcrx} &= \mathbf{R}_{11}^o, & \text{oz} &= \psi^o
\end{aligned}$$

5.5 BasicLaserOdometry::updateIMU()

ScanRegistration ノードから/imu_trans トピックを通じて受け取った, IMU に関連するデータを BasicLaserOdometry の各メンバに格納する. オイラー角に関しては, ROS の座標軸における $\tilde{X}, \tilde{Y}, \tilde{Z}$ 軸周りの回転を表す. また, $k+1$ 番目の最新の点群 (スキャン開始時刻 t_{k+1} , 点群のスキャン間隔_scanPeriod) と対応する. 時刻 t_{k+1} の IMU 座標系を S , 時刻 t_{k+1} から $t_{k+2} = t_{k+1} + \text{_scanPeriod}$ までの間定速移動していた場合の IMU 座標系を S' , 時刻 t_{k+2} の IMU 座標系を C , グローバル座標系を G とする. 座標系 S と S' は互いに平行移動の関係にあるので, G から S への座標系の回転と, G から S' への座標系の回転は等しくなる.

- _imuPitchStart, _imuYawStart, _imuRollStart: G から S , あるいは G から S' への座標系の回転. 即ち, 点の座標を S から G , あるいは S' から G へと移すような回転.
- _imuPitchEnd, _imuYawEnd, _imuRollEnd: G から C への座標系の回転. 即ち, 点の座標を C から G へと移すような回転.
- _imuShiftFromStart: 時刻 t_{k+1} から t_{k+2} にかけての速度変化を考慮するための補正量 (S あるいは S' 座標系).
- _imuVeloFromStart: 時刻 t_{k+1} から t_{k+2} にかけての速度変化 (S あるいは S' 座標系).

5.6 BasicLaserOdometry::process()

5.6.1 概要

LaserOdometry ノードの主な処理を行う. 即ち, 最新 ($k+1$ 番目) の点群 \mathcal{P}_{k+1} を, 直前 (k 番目) の点群 $\overline{\mathcal{P}}_k$ にレジストレーション (重ね合わせ) することで, 時刻 t_{k+1} から次の時刻 t_{k+2} にかけての LiDAR の姿勢変化 T_{k+1}^L を計算する. 点群 \mathcal{P}_{k+1} および $\overline{\mathcal{P}}_k$ に含まれる各点の座標は, IMU データを用いて, 点群 \mathcal{P}_{k+1} のスキャン開始時刻 ($k+1$ 番目のスイープ開始時刻) t_{k+1} に揃えられている. 点群の取得間隔は_scanPeriod であり, 以後簡単のため Δt とおく (デフォルトでは 100ms). 次 ($k+2$ 番目) の点群のスキャン開始時刻 t_{k+2} は, $t_{k+2} = t_{k+1} + \text{_scanPeriod} = t_{k+1} + \Delta t$ のように表せる. 最新の点群 \mathcal{P}_{k+1} には, 時刻 t_{k+1} から t_{k+2} までの間に取得された点が含まれる. 同様に, 直前の点群 $\overline{\mathcal{P}}_k$ には, 時刻 t_k から t_{k+1} までの間に取得された点が含まれる. このメソッドは長大である (470 行を超える) ので, 分割して説明する.

5.6.2 コーナーの対応付け

直前 (k 番目) の点群 $\overline{\mathcal{P}}_k$ に含まれる点のうち, それほど顕著ではないコーナー, あるいは顕著なコーナーとしてラベル付けされた点のセットが_lastCornerCloud に格納されている (これを $\overline{\mathcal{A}}_k$ とする). 現在 ($k+1$ 番目) の点群 \mathcal{P}_{k+1} に含まれる点のうち, 顕著なコーナーとしてラベル付けされた点のセットが_cornerPointsSharp に格納されている (論文の \mathcal{E}_{k+1}). lastCornerCloudSize と cornerPointsSharpNum はそれぞれ, $\overline{\mathcal{A}}_k, \mathcal{E}_{k+1}$ に含まれる点の個数である. \mathcal{E}_{k+1} に含まれる各点に対して, それと対応する点を $\overline{\mathcal{A}}_k$ から探索する. 下記の対応付けは, 論文の手法 (Section V-B) に概ね従っている.

\mathcal{E}_{k+1} 内の点 i の座標を $X_{(k+1,i)}^L$ とおく. 座標 $X_{(k+1,i)}^L$ は IMU データによってある程度, 時刻 t_{k+1} に揃えられている (即ち, 時刻 t_{k+1} の LiDAR 座標系での座標にある程度近づけられている). transformToStart()

を呼び出すことで、現在の姿勢の解 T_{k+1}^L を使って、点 i の座標 $X_{(k+1,i)}^L$ を、時刻 t_{k+1} の LiDAR 座標系における座標 $\widetilde{X}_{(k+1,i)}^L$ に変換する。即ち座標 $X_{(k+1,i)}^L$ は、IMU データと姿勢 T_{k+1}^L の双方を使って、時刻 t_{k+1} に揃えられる (IMU データを使って大まかに揃えた後、姿勢の推定値を使って精度良く揃える)。 \mathcal{E}_{k+1} 内の各点の座標を、現在の姿勢の解 T_{k+1}^L を使って時刻 t_{k+1} に揃えることで得た点群を $\widetilde{\mathcal{E}}_{k+1}$ とする (論文と同様)。

$\widetilde{\mathcal{E}}_{k+1}$ 内の点 i (座標は $\widetilde{X}_{(k+1,i)}^L$) について、最近傍点を $\overline{\mathcal{A}}_k$ の中から探索する。 $\widetilde{X}_{(k+1,i)}^L$ は pointSel に対応する。_lastCornerCloud から構築された KD 木 (_lastCornerKdTree) を使って効率良く探索する。最近傍点のインデックスが closestPointInd, 点 $i \in \widetilde{\mathcal{E}}_{k+1}$ から最近傍点までの二乗距離が pointSearchSqDis[0] に格納される。論文に倣って、この最近傍点を $j \in \overline{\mathcal{A}}_k$, 点 j の座標を $\overline{X}_{(k,j)}^L$ で表す。座標 $\overline{X}_{(k,j)}^L$ は、時刻 t_{k+1} の LiDAR 座標系で表現される。また $\overline{\mathcal{A}}_k$ は $\overline{\mathcal{P}}_k$ の部分集合ゆえ $j \in \overline{\mathcal{P}}_k$ である。インデックス closestPointInd は j , 二乗距離 pointSearchSqDis[0] は $\|\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,j)}^L\|^2$ と書ける。点 $i \in \widetilde{\mathcal{E}}_{k+1}$ と点 $j \in \overline{\mathcal{A}}_k$ との二乗距離が 25 以下 (距離が 5m 未満) であれば、点 i と j は対応するものとして扱う。距離が 5m 以上の場合は、点 i の最近傍点 j が点群 $\overline{\mathcal{A}}_k$ に存在しないことを意味するから、インデックス closestPointInd には -1 を格納する。

点 j が含まれるリングの番号 (スキャン ID) を intensity フィールドの整数部分から得て、closestPointScan に格納する (ScanRegistration の説明を参照)。リングとは、同一の仰角 (Vertical Angle) をもつ点の集合である (図 1)。点 j を含むリング (closestPointScan) の周囲のリング (closestPointScan - 2 から closestPointScan + 2 まで、但し closestPointScan は除く) から、点 i の最近傍点を探索する。最近傍点のインデックスが minPointInd2, 点 i から最近傍点までの二乗距離が minPointSqDis2 に格納される。論文に倣って、この最近傍点を $l \in \overline{\mathcal{A}}_k$, 点 l の座標を $\overline{X}_{(k,l)}^L$ で表す。点 j と l はいずれも点群 $\overline{\mathcal{A}}_k$ に含まれるが、 l は j とは異なる (closestPointScan ではない) リングに属する。インデックス minPointInd2 は l , 二乗距離 minPointSqDis2 は $\|\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,l)}^L\|^2$ と書ける。点 $i \in \widetilde{\mathcal{E}}_{k+1}$ と点 $l \in \overline{\mathcal{A}}_k$ との二乗距離が 25 以下 (距離が 5m 未満) であれば、点 i と l は対応するものとして扱う。距離が 5m 以上の場合は、条件を満たすような最近傍点 l が点群 $\overline{\mathcal{A}}_k$ に存在しないことを意味するから、インデックス minPointInd2 には -1 を格納する。最近傍点 j が存在しなければ (closestPointInd が -1 であれば)、最近傍点 l も存在しないから minPointInd2 は -1 になる。点 i に対応する _pointSearchCornerInd1 と _pointSearchCornerInd2 の要素には、点 j および l のインデックス closestPointInd, minPointInd2 を格納し、インデックスの対応関係 i, j, l を記憶しておく。

5.6.3 対応する点とエッジ間の距離計算

点 $i \in \widetilde{\mathcal{E}}_{k+1}$ の最近傍点 $j, l \in \overline{\mathcal{A}}_k$ のインデックスは、_pointSearchCornerInd1 と _pointSearchCornerInd2 から参照できる (先程の closestPointInd と minPointInd2 に対応する)。双方のインデックスが有効であれば (-1 でなければ)、以下の処理を続ける。

点 j と l は、コーナーを集めた点群 $\overline{\mathcal{A}}_k$ に属するので、点 j と l を結んだ線はエッジを表している。従って点 i, j, l は、 $\overline{\mathcal{A}}_k$ におけるエッジ (論文の用語では Edge line) と、 $\widetilde{\mathcal{E}}_{k+1}$ に含まれる点との対応関係を表している。両者の距離 $d_{\mathcal{E}}$ を最小化することで、2 つの点群 $\overline{\mathcal{A}}_k$ と $\widetilde{\mathcal{E}}_{k+1}$, 言い換えると $\overline{\mathcal{P}}_k$ と \mathcal{P}_{k+1} を重ね合わせることで、LaserOdometry ノードの当初の目的を実現できる。論文の式 (2) にあるように、点と直線間の距離 $d_{\mathcal{E}}$ は次のように記述される。

$$d_{\mathcal{E}} = \frac{\left\| \left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,j)}^L \right) \times \left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,l)}^L \right) \right\|}{\left\| \overline{X}_{(k,j)}^L - \overline{X}_{(k,l)}^L \right\|} \quad (20)$$

点 i, j, l の座標 $\widetilde{\mathbf{X}}_{(k+1,i)}^L, \overline{\mathbf{X}}_{(k,j)}^L, \overline{\mathbf{X}}_{(k,l)}^L$ は, pointSel, tripod1, tripod2 に対応する. これらの座標から (20) 式の分子と分母を求めて, a012 と 112 に格納する. a012 を 112 で割れば (20) 式の距離 $d_{\mathcal{E}}$ が得られ, これを 1d2 に格納する. 座標を $\widetilde{\mathbf{X}}_{(k+1,i)}^L = [x_0, y_0, z_0]^\top$, $\overline{\mathbf{X}}_{(k,j)}^L = [x_1, y_1, z_1]^\top$, $\overline{\mathbf{X}}_{(k,l)}^L = [x_2, y_2, z_2]^\top$ とおいて, (20) 式の分子を展開してみると次のようになる.

$$\begin{aligned}
& \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,j)}^L \right) \times \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \\
&= [x_0 - x_1, y_0 - y_1, z_0 - z_1]^\top \times [x_0 - x_2, y_0 - y_2, z_0 - z_2]^\top \\
&= [(y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2), \\
&\quad (z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2), \\
&\quad (x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)]^\top
\end{aligned} \tag{21}$$

$\widetilde{\mathbf{X}}_{(k+1,i)}^L = [x_0, y_0, z_0]^\top$ は変数 x_0, y_0, z_0 , $\overline{\mathbf{X}}_{(k,j)}^L = [x_1, y_1, z_1]^\top$ は変数 x_1, y_1, z_1 , $\overline{\mathbf{X}}_{(k,l)}^L = [x_2, y_2, z_2]^\top$ は変数 x_2, y_2, z_2 に対応する. 変数 a012 は以下の計算によって得られる.

$$\begin{aligned}
& \left\| \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,j)}^L \right) \times \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \right\| \\
&= \left(((y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2))^2 \right. \\
&\quad \left. + ((z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2))^2 \right. \\
&\quad \left. + ((x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2))^2 \right)^{\frac{1}{2}}
\end{aligned} \tag{22}$$

(20) 式の方母 (変数 112) は次のようになる.

$$\left\| \overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right\| = \left((x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \right)^{\frac{1}{2}} \tag{23}$$

(20) 式の距離 $d_{\mathcal{E}}$ だけでなく, 後ほどの最適化処理で用いるために, 以下のベクトル $[l_a, l_b, l_c]^\top$ も計算しておく, その各要素を $1a, 1b, 1c$ に格納する.

$$[l_a, l_b, l_c]^\top = \frac{\left(\overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \times \left(\left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,j)}^L \right) \times \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \right)}{\left\| \overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right\| \left\| \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,j)}^L \right) \times \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \right\|} \tag{24}$$

(24) 式の方母は, a012 および 112 の積である. また (24) 式の方母を展開してみると次のようになる.

$$\begin{aligned}
& \left(\overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \times \left(\left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,j)}^L \right) \times \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \right) \\
&= [x_1 - x_2, y_1 - y_2, z_1 - z_2]^\top \\
&\quad \times [(y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2), \\
&\quad (z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2), \\
&\quad (x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)]^\top \\
&= [(y_1 - y_2)((x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)) \\
&\quad - (z_1 - z_2)((z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2)), \\
&\quad (z_1 - z_2)((y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2)) \\
&\quad - (x_1 - x_2)((x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)), \\
&\quad (x_1 - x_2)((z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2)) \\
&\quad - (y_1 - y_2)((y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2))]^\top
\end{aligned}$$

さて、ベクトル a, b の外積 $a \times b$ と、ノルム $\|a\|, \|b\|$ との間には $\|a\|\|b\| \sin \theta = \|a \times b\|$ の関係が成り立つ ($0 \leq \theta \leq \pi$ は a と b のなす角). また内積 $a^\top b$ と、ノルム $\|a\|, \|b\|$ との間には $\|a\|\|b\| \cos \theta = a^\top b$ の関係が成り立つ. 2つのベクトルが直交していれば ($\theta = \pi/2$), $\cos \theta = 0, \sin \theta = 1$ であるから, $\|a\|\|b\| = \|a \times b\|$ および $a^\top b = 0$ が成立する. ベクトル $b - c$ と $(a - b) \times (a - c)$ は、内積が0になるから互いに直交する.

$$\begin{aligned}
& (b - c)^\top ((a - b) \times (a - c)) \\
&= (b - c)^\top (a \times a - a \times c - b \times a + b \times c) \\
&= (b - c)^\top (-a \times c + a \times b + b \times c) \\
&= (b - c)^\top (a \times (b - c) + b \times c) \\
&= (b - c)^\top (a \times (b - c)) + (b - c)^\top (b \times c) \\
&= a^\top ((b - c) \times (b - c)) + b^\top (b \times c) - c^\top (b \times c) \\
&= c^\top (b \times b) - b^\top (c \times c) \\
&= 0
\end{aligned}$$

上式の変形では、外積に関する次の関係を用いている.

$$\begin{aligned}
a \times a &= 0 \\
a \times b &= -b \times a \\
a \times (b + c) &= a \times b + a \times c \\
a^\top (b \times c) &= b^\top (c \times a) = c^\top (a \times b)
\end{aligned}$$

$b - c$ と $(a - b) \times (a - c)$ が直交するので、以下が成立する.

$$\|b - c\| \|(a - b) \times (a - c)\| = \|(b - c) \times ((a - b) \times (a - c))\|$$

$a = \widetilde{X}_{(k+1,i)}^L, b = \overline{X}_{(k,j)}^L, c = \overline{X}_{(k,l)}^L$ と置き換えれば以下のようになる.

$$\begin{aligned}
& \left\| \overline{X}_{(k,j)}^L - \overline{X}_{(k,l)}^L \right\| \left\| \left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,j)}^L \right) \times \left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,l)}^L \right) \right\| \\
&= \left\| \left(\overline{X}_{(k,j)}^L - \overline{X}_{(k,l)}^L \right) \times \left(\left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,j)}^L \right) \times \left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,l)}^L \right) \right) \right\|
\end{aligned}$$

ベクトル $[l_a, l_b, l_c]^\top$ は結局次のように書けるから、ノルムが1に正規化された単位ベクトルであることが確かめられる.

$$[l_a, l_b, l_c]^\top = \frac{\left(\overline{X}_{(k,j)}^L - \overline{X}_{(k,l)}^L \right) \times \left(\left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,j)}^L \right) \times \left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,l)}^L \right) \right)}{\left\| \left(\overline{X}_{(k,j)}^L - \overline{X}_{(k,l)}^L \right) \times \left(\left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,j)}^L \right) \times \left(\widetilde{X}_{(k+1,i)}^L - \overline{X}_{(k,l)}^L \right) \right) \right\|} \quad (25)$$

ベクトル $a \times b$ は a および b に直交することを考えると ($a^\top (a \times b) = 0, b^\top (a \times b) = 0$), 上記のベクトル $[l_a, l_b, l_c]^\top$ は $\overline{X}_{(k,j)}^L - \overline{X}_{(k,l)}^L$ に直交する. これは、ベクトル $[l_a, l_b, l_c]^\top$ の方向が、点 j, l を結ぶエッジ (直線) に対して垂直となることを意味するから、ベクトル $[l_a, l_b, l_c]^\top$ はエッジの法線ベクトルでもある. 以上をまとめれば、変数 $1a, 1b, 1c$ には、点 j, l を結ぶエッジの単位法線ベクトルを、変数 $1d2$ には点 i とエッジとの距離を格納している. この4つを変数 `coeff` にセットしている.

論文 (Section V-D) で言及されている通り、点と直線との距離が大きいかほど、点 i, j, l の対応関係は信頼できないものとなる. 距離が大きく信頼できない対応関係には小さな重みを与えて、外れ値による影響を軽減する. 外れ値による影響とは、点群間の誤った対応付けによって、極端に距離の大きな対応関係が作り出され、最終的

な解がこの対応関係 (外れ値) に引っ張られる (最適解から外れる) ことを指す。各対応関係に対する重み付けの係数は変数 s に格納され, coeff の各要素 (距離と単位法線ベクトル) は s によってスケールされる。現在の姿勢の解 T_{k+1}^L を使って変換する前の, 元々の点 $i \in \mathcal{E}_{k+1}$ の座標 $X_{(k+1,i)}^L$ を $_laserCloudOri$ に, また上記の coeff (スケールされた距離と単位法線ベクトルのセット) を $_coeffSel$ にそれぞれ追加する。これらは後ほどの最適化処理で使用される。

5.6.4 平面の対応付け

平面の対応付けは, コーナーの対応付けと同じ流れで行うので, 説明も似たようになる。直前 (k 番目) の点群 $\bar{\mathcal{P}}_k$ に含まれる点のうち, それほど顕著ではない平面, あるいは顕著な平面としてラベル付けされた点のセットが $_lastSurfaceCloud$ に格納されている (これを $\bar{\mathcal{B}}_k$ とする)。現在 ($k+1$ 番目) の点群 \mathcal{P}_{k+1} に含まれる点のうち, 顕著な平面としてラベル付けされた点のセットが $_surfPointsFlat$ に格納されている (論文の \mathcal{H}_{k+1})。 $_lastSurfaceCloudSize$ と $_surfPointsFlatNum$ はそれぞれ, $\bar{\mathcal{B}}_k, \mathcal{H}_{k+1}$ に含まれる点の個数である。 \mathcal{H}_{k+1} に含まれる各点に対して, それと対応する点を $\bar{\mathcal{B}}_k$ から探索する。下記の対応付けは, 論文の手法 (Section V-B) に概ね従っている。

\mathcal{H}_{k+1} 内の点 i の座標を $X_{(k+1,i)}^L$ とおく。座標 $X_{(k+1,i)}^L$ は IMU データによってある程度, 時刻 t_{k+1} に揃えられている。コーナー \mathcal{E}_{k+1} の場合と同様に $\text{transformToStart}()$ を呼び出すことで, 現在の姿勢の解 T_{k+1}^L を使って, 点 i の座標 $X_{(k+1,i)}^L$ を, 時刻 t_{k+1} の LiDAR 座標系における座標 $\tilde{X}_{(k+1,i)}^L$ に変換する。即ち座標 $X_{(k+1,i)}^L$ は, IMU データと姿勢 T_{k+1}^L の双方を使って, 時刻 t_{k+1} に揃えられる。 \mathcal{H}_{k+1} 内の各点の座標を, 現在の姿勢の解 T_{k+1}^L を使って時刻 t_{k+1} に揃えることで得た点群を $\tilde{\mathcal{H}}_{k+1}$ とする (論文と同様)。

先程と同様に, $\tilde{\mathcal{H}}_{k+1}$ 内の点 i (座標は $X_{(k+1,i)}^L$) について, 最近傍点を $\bar{\mathcal{B}}_k$ の中から探索する。 $\tilde{X}_{(k+1,i)}^L$ は pointSel に対応する。 $_lastSurfaceCloud$ から構築された KD 木 ($_lastSurfaceKDTree$) を使って探索の効率を上げる。最近傍点のインデックスが closestPointInd , 点 $i \in \tilde{\mathcal{H}}_{k+1}$ から最近傍点までの二乗距離が $\text{pointSearchSqDis}[0]$ に格納される。この最近傍点を $j \in \bar{\mathcal{B}}_k$, 点 j の座標を $\bar{X}_{(k,j)}^L$ で表す ($\bar{X}_{(k,j)}^L$ は時刻 t_{k+1} の LiDAR 座標系)。 $\bar{\mathcal{B}}_k$ は $\bar{\mathcal{P}}_k$ の部分集合であるから $j \in \bar{\mathcal{P}}_k$ である。インデックス closestPointInd は j , 二乗距離 $\text{pointSearchSqDis}[0]$ は $\|\tilde{X}_{(k+1,i)}^L - \bar{X}_{(k,j)}^L\|^2$ である。点 $i \in \tilde{\mathcal{H}}_{k+1}$ と点 $j \in \bar{\mathcal{B}}_k$ との距離が 5m 未満であれば, 点 i と j は対応するものとして扱う。距離が 5m 以上の場合は, 点 i の最近傍点 j が点群 $\bar{\mathcal{B}}_k$ に存在しないことを意味するから, インデックス closestPointInd には -1 を格納する。

ここからの処理はコーナーの対応付けとは異なる。点 j が含まれるリングの番号を closestPointScan に格納する。続いて, 点 j と同一のリング closestPointScan から, 点 i と 2 番目に近い点を探索する。この点のインデックスが minPointInd2 , 点 i からこの点までの二乗距離が minPointSqDis2 に格納される。論文に倣って, 点 j に次ぐ最近傍点を $l \in \bar{\mathcal{B}}_k$, 点 l の座標を $\bar{X}_{(k,l)}^L$ で表す。リング closestPointScan のなかで, 点 i と最も近い点が j , 2 番目に近い点が l であるから, 明らかに $j \neq l$ である。インデックス l は minPointInd2 , 二乗距離 minPointSqDis2 は $\|\tilde{X}_{(k+1,i)}^L - \bar{X}_{(k,l)}^L\|^2$ と書ける。更に, closestPointScan の周囲のリング ($\text{closestPointScan} - 2$ から $\text{closestPointScan} + 2$ まで, 但し closestPointScan を除く) から, 点 i の最近傍点を探索する。この点は j とは異なるリングに含まれており, そのインデックスは minPointInd3 , 点 i からこの点までの二乗距離が minPointSqDis3 に格納される。論文に倣って, 最近傍点のインデックスを $m \in \bar{\mathcal{B}}_k$, 点 m の座標を $\bar{X}_{(k,m)}^L$ で表す。インデックス m は minPointInd3 , 二乗距離 minPointSqDis3 は $\|\tilde{X}_{(k+1,i)}^L - \bar{X}_{(k,m)}^L\|^2$ のように表される。

点 $i \in \tilde{\mathcal{H}}_{k+1}$ と $l \in \bar{\mathcal{B}}_k$ との距離が 5m 未満であれば, 点 i と l は妥当な対応関係として扱う。もし距離

が 5m 以上であれば、条件を満たすような点 l が点群 \bar{B}_k 内に存在しないことを意味するから、インデックス minPointInd2 には -1 をセットする。点 $i \in \tilde{\mathcal{H}}_{k+1}$ と $m \in \bar{B}_k$ についても同様で、両者の距離が 5m 未満ならば有効な対応関係として扱い、5m 以上であればインデックス minPointInd3 には -1 をセットする (条件を満たすような点 m が点群 \bar{B}_k 内に存在しない)。最近傍点 j が存在しなければ (closestPointInd が -1 であれば)、点 l, m の探索も行われなから minPointInd2 , minPointInd3 は -1 になる。点 i に対応する $_pointSearchCornerInd1$, $_pointSearchCornerInd2$, $_pointSearchCornerInd3$ の要素には、点 j, l, m のインデックス closestPointInd , minPointInd2 , minPointInd3 を格納し、インデックスの対応関係 i, j, l, m を記憶しておく。

5.6.5 点と平面間の距離計算

点 $i \in \tilde{\mathcal{H}}_{k+1}$ の最近傍点 $j, l, m \in \bar{B}_k$ のインデックスは、 $_pointSearchCornerInd1$, $_pointSearchCornerInd2$, $_pointSearchCornerInd3$ から参照できる (先程の closestPointInd , minPointInd2 , minPointInd3 に対応する)。3 つのインデックスが有効であれば (-1 でなければ)、以下の処理を続ける。

点 j, l, m は、 $\bar{\mathcal{P}}_k$ のなかから平面として分類された点を集めた点群 \bar{B}_k に属し、かつ点 i の近傍であるから、点 j, l, m は同一の平面を表している。従って点 i, j, l, m は、 \bar{B}_k における平面 (論文の用語では Planar patch) と、 $\tilde{\mathcal{H}}_{k+1}$ に含まれる点との対応関係を表している。両者の距離 $d_{\mathcal{H}}$ を最小化すれば、2 つの点群 $\bar{\mathcal{P}}_k$ と \mathcal{P}_{k+1} が重なり合う。論文の式 (3) にあるように、点と平面間の距離 $d_{\mathcal{H}}$ は次のように記述される。

$$d_{\mathcal{H}} = \frac{\left(\tilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,j)}^L \right)^\top \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right)}{\left\| \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right\|} \quad (26)$$

論文では、距離を正にするために (26) の分子の絶対値をとっているが、実装では絶対値をとらない。後続の最適化では、距離の二乗 $d_{\mathcal{H}}^2$ を最小化するように姿勢 T_{k+1}^L が更新されるので、距離 $d_{\mathcal{H}}$ の符号は負でも構わない。点 i, j, l, m の座標 $\tilde{\mathbf{X}}_{(k+1,i)}^L, \bar{\mathbf{X}}_{(k,j)}^L, \bar{\mathbf{X}}_{(k,l)}^L, \bar{\mathbf{X}}_{(k,m)}^L$ は、 pointSel , tripod1 , tripod2 , tripod3 に対応する。座標を $\tilde{\mathbf{X}}_{(k+1,i)}^L = [x_0, y_0, z_0]^\top$, $\bar{\mathbf{X}}_{(k,j)}^L = [x_1, y_1, z_1]^\top$, $\bar{\mathbf{X}}_{(k,l)}^L = [x_2, y_2, z_2]^\top$, $\bar{\mathbf{X}}_{(k,m)}^L = [x_3, y_3, z_3]^\top$ とおく。最初に以下の外積を求めて、その各要素を pa , pb , pc に格納する。

$$\begin{aligned} & \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \\ &= [x_1 - x_2, y_1 - y_2, z_1 - z_2]^\top \times [x_1 - x_3, y_1 - y_3, z_1 - z_3]^\top \\ &= [(y_1 - y_2)(z_1 - z_3) - (z_1 - z_2)(y_1 - y_3), \\ & \quad (z_1 - z_2)(x_1 - x_3) - (x_1 - x_2)(z_1 - z_3), \\ & \quad (x_1 - x_2)(y_1 - y_3) - (y_1 - y_2)(x_1 - x_3)]^\top \end{aligned} \quad (27)$$

次に以下の内積を求めて、 pd に格納する ((26) 式の分子を展開して得られる第 2 項である)。 pa , pb , pc , そして tripod1 を用いる。

$$\begin{aligned} & - \left(\bar{\mathbf{X}}_{(k,j)}^L \right)^\top \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right) \\ &= - (x_1 ((y_1 - y_2)(z_1 - z_3) - (z_1 - z_2)(y_1 - y_3)) \\ & \quad + x_2 ((z_1 - z_2)(x_1 - x_3) - (x_1 - x_2)(z_1 - z_3)) \\ & \quad + x_3 ((x_1 - x_2)(y_1 - y_3) - (y_1 - y_2)(x_1 - x_3))) \end{aligned} \quad (28)$$

変数 p_s は先ほど外積のノルムであり、以下の計算によって得られる。 p_a, p_b, p_c を用いる。

$$\begin{aligned}
& \left\| \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right\| \\
&= \left(((y_1 - y_2)(z_1 - z_3) - (z_1 - z_2)(y_1 - y_3))^2 \right. \\
&\quad + ((z_1 - z_2)(x_1 - x_3) - (x_1 - x_2)(z_1 - z_3))^2 \\
&\quad \left. + ((x_1 - x_2)(y_1 - y_3) - (y_1 - y_2)(x_1 - x_3))^2 \right)^{\frac{1}{2}} \tag{29}
\end{aligned}$$

この時点での p_a, p_b, p_c, p_d, p_s を p_a, p_b, p_c, p_d, p_s とする。 p_a, p_b, p_c, p_d を p_s で割った後、(26) 式の距離 $d_{\mathcal{H}}$ を次のように求める。

$$\begin{aligned}
d_{\mathcal{H}} &= \frac{1}{p_s} (p_a x_0 + p_b y_0 + p_c z_0 + p_d) \\
&= \frac{1}{p_s} \left([x_0, y_0, z_0] [p_a, p_b, p_c]^{\top} + p_d \right) \\
&= \left\| \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right\|^{-1} \\
&\quad \left(\left(\tilde{\mathbf{X}}_{(k+1,i)}^L \right)^{\top} \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right) \right. \\
&\quad \left. - \left(\bar{\mathbf{X}}_{(k,j)}^L \right)^{\top} \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right) \right) \tag{30}
\end{aligned}$$

(26) 式から計算した距離 $d_{\mathcal{H}}$ だけでなく、後の最適化ではベクトル $[p_a/p_s, p_b/p_s, p_c/p_s]^{\top}$ も使用する。

$$\frac{1}{p_s} [p_a, p_b, p_c]^{\top} = \frac{\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right)}{\left\| \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right\|} \tag{31}$$

ベクトル $[p_a, p_b, p_c]^{\top}$ は、平面上の点 j と l 、点 j と m を結ぶ2本のベクトル $\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L, \bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L$ に垂直であるから、平面に対する法線ベクトルとなっている。そして、各要素をノルム p_s で割って得られる $[p_a/p_s, p_b/p_s, p_c/p_s]^{\top}$ は単位ベクトルである。従って先程と同様に、距離と単位法線ベクトルを求めて、変数 `coeff` にセットしている。対応関係 i, j, l, m に対する重み付けの係数は変数 `s` に格納され、`coeff` の各要素は `s` によりスケールされる。現在の姿勢 T_{k+1}^L を使って変換する前の、元々の点 $i \in \mathcal{H}_{k+1}$ の座標 $X_{(k+1,i)}^L$ を `laserCloud0ri` に、また上記の `coeff`(スケールされた距離と単位法線ベクトルのセット) を `_coeffSel` にそれぞれ追加する。これらは後ほどの最適化処理で使用される。

5.6.6 最適化による姿勢の更新

上記の対応付け処理の後には、最適化処理、具体的にはガウス・ニュートン法による姿勢 T_{k+1}^L の更新が控えている。ガウス・ニュートン法では、点とエッジ間の距離 $d_{\mathcal{E}}$ 、あるいは点と平面間の距離 $d_{\mathcal{H}}$ を最小化するような姿勢 T_{k+1}^L を逐次的に求める。姿勢 T_{k+1}^L は、最新の点群 \mathcal{P}_{k+1} に含まれるコーナーあるいは平面上の点 i の座標 $X_{(k+1,i)}^L$ と、時刻 t_{k+1} の LiDAR 座標系での座標 $\tilde{X}_{(k+1,i)}^L$ との関係を表す。座標 $X_{(k+1,i)}^L$ は、IMU データによってある程度補正され、時刻 t_{k+1} に揃えられている(時刻 t_{k+1} の LiDAR 座標系での座標にある程度近づけられている)場合もあるが、IMU を使っていなければ座標は補正されないで、点 i が取得された時点の LiDAR 座標系で表されている。ここでは後者の場合、即ち、点 i の座標 $X_{(k+1,i)}^L$ は点 i が取得された時点(時刻 t_i とする)の LiDAR 座標系で表されるとする。時刻 t_i は、最初の点であれば点群 \mathcal{P}_{k+1} のスキャ

ン開始時刻 t_{k+1} に等しく、最後の点であれば次の点群 \mathcal{P}_{k+2} のスキャン開始時刻 t_{k+2} となる (点群 \mathcal{P}_{k+1} は、時刻 t_{k+1} から t_{k+2} までの間に取得された点を含んでおり、点群の取得間隔を Δt とすれば $t_{k+2} = t_{k+1} + \Delta t$ が成り立つ)。

`transformToStart()` の (14) 式に示した通り、姿勢 $\mathbf{T}_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$ は、時刻 t_{k+1} の LiDAR 座標系での座標 \mathbf{p} を、時刻 t_{k+2} の LiDAR 座標系での座標 \mathbf{p}' に変換する。

$$\begin{aligned}\mathbf{p}' &= \mathbf{R}(-\theta_x, -\theta_y, -\theta_z)^\top \mathbf{p} - [t_x, t_y, t_z]^\top \\ &= (\mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z))^\top \mathbf{p} + [t_x, t_y, t_z]^\top \\ &= (\mathbf{R}_z(\theta_z)\mathbf{R}_x(\theta_x)\mathbf{R}_y(\theta_y)) \mathbf{p} + [t_x, t_y, t_z]^\top\end{aligned}$$

LOAM では、1 つの点群を取得する間 (時刻 t_{k+1} から t_{k+2} までの間) は定速移動だと仮定される。この仮定のもとでは、時刻 $t(t_{k+1} \leq t \leq t_{k+2})$ の LiDAR 座標系における点の座標 $\mathbf{p}(t)$ と \mathbf{p} との間に次の関係が成り立つ ($\Delta t = t_{k+2} - t_{k+1}$)。

$$\begin{aligned}\mathbf{p}(t) &= \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z)^\top \mathbf{p} + [st_x, st_y, st_z]^\top \\ &= (\mathbf{R}_y(-s\theta_y)\mathbf{R}_x(-s\theta_x)\mathbf{R}_z(-s\theta_z))^\top \mathbf{p} + [st_x, st_y, st_z]^\top, \quad s = \frac{t - t_{k+1}}{\Delta t}\end{aligned}$$

上式を使えば、時刻 $t_i(t_{k+1} \leq t_i \leq t_{k+2})$ で取得された点 i の座標 $\mathbf{X}_{(k+1,i)}^L$ と、時刻 t_{k+1} の LiDAR 座標系における点 i の座標 $\widetilde{\mathbf{X}}_{(k+1,i)}^L$ との関係式が得られる (論文の式 (5))。

$$\mathbf{X}_{(k+1,i)}^L = (\mathbf{R}_y(-s\theta_y)\mathbf{R}_x(-s\theta_x)\mathbf{R}_z(-s\theta_z))^\top \widetilde{\mathbf{X}}_{(k+1,i)}^L + [st_x, st_y, st_z]^\top \quad (32)$$

$$\begin{aligned}&= (\mathbf{R}_z(-s\theta_z)^\top \mathbf{R}_x(-s\theta_x)^\top \mathbf{R}_y(-s\theta_y)^\top) \widetilde{\mathbf{X}}_{(k+1,i)}^L + [st_x, st_y, st_z]^\top \\ &= (\mathbf{R}_z(s\theta_z)\mathbf{R}_x(s\theta_x)\mathbf{R}_y(s\theta_y)) \widetilde{\mathbf{X}}_{(k+1,i)}^L + [st_x, st_y, st_z]^\top, \quad s = \frac{t_i - t_{k+1}}{\Delta t}\end{aligned} \quad (33)$$

これを $\widetilde{\mathbf{X}}_{(k+1,i)}^L$ についての式に直すと次のようになる。

$$\widetilde{\mathbf{X}}_{(k+1,i)}^L = (\mathbf{R}_y(-s\theta_y)\mathbf{R}_x(-s\theta_x)\mathbf{R}_z(-s\theta_z)) \left(\mathbf{X}_{(k+1,i)}^L - [st_x, st_y, st_z]^\top \right) \quad (34)$$

$$= \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z) \left(\mathbf{X}_{(k+1,i)}^L - [st_x, st_y, st_z]^\top \right) \quad (35)$$

$\mathbf{X}_{(k+1,i)}^L$ は \mathcal{E}_{k+1} あるいは \mathcal{H}_{k+1} に含まれる点 i の座標、 $\widetilde{\mathbf{X}}_{(k+1,i)}^L$ は $\widetilde{\mathcal{E}}_{k+1}$ あるいは $\widetilde{\mathcal{H}}_{k+1}$ に含まれる点 i の座標である。(35) 式を (20) 式と (26) 式に代入することで、距離 $d_{\mathcal{E}}$ と $d_{\mathcal{H}}$ を \mathbf{T}_{k+1}^L の関数 $f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ 、 $f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ として書き直せる (論文の式 (9) と (10))。

$$\begin{aligned}f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) &= \left\| \overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right\|^{-1} \\ &\quad \left\| \left(\mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z) \left(\mathbf{X}_{(k+1,i)}^L - [st_x, st_y, st_z]^\top \right) - \overline{\mathbf{X}}_{(k,j)}^L \right) \right. \\ &\quad \left. \times \left(\mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z) \left(\mathbf{X}_{(k+1,i)}^L - [st_x, st_y, st_z]^\top \right) - \overline{\mathbf{X}}_{(k,l)}^L \right) \right\| \quad (36)\end{aligned}$$

$$\begin{aligned}f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) &= \left\| \left(\overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \times \left(\overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,m)}^L \right) \right\|^{-1} \\ &\quad \left(\mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z) \left(\mathbf{X}_{(k+1,i)}^L - [st_x, st_y, st_z]^\top \right) - \overline{\mathbf{X}}_{(k,j)}^L \right)^\top \\ &\quad \left(\left(\overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,l)}^L \right) \times \left(\overline{\mathbf{X}}_{(k,j)}^L - \overline{\mathbf{X}}_{(k,m)}^L \right) \right) \quad (37)\end{aligned}$$

ガウス・ニュートン法を使って、距離 $d_{\mathcal{E}} = f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ と $d_{\mathcal{H}} = f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ を \mathbf{T}_{k+1}^L に関して最小化したいが、そのためには $f_{\mathcal{E}}$ と $f_{\mathcal{H}}$ の、 \mathbf{T}_{k+1}^L に関する勾配ベクトル (微分) $\nabla f_{\mathcal{E}}, \nabla f_{\mathcal{H}}$ が必要である。スカラー関数 $f_{\mathcal{E}}, f_{\mathcal{H}}$ の 6 次元ベクトル \mathbf{T}_{k+1}^L による微分だから、 $\nabla f_{\mathcal{E}}, \nabla f_{\mathcal{H}}$ も 6 次元ベクトルとなる。いきなり微分を計算すると大変なので、微分を求めやすくするための関係式を予め導いておく。

x の関数で表されるベクトル $\mathbf{v}(x) = [f(x), g(x), h(x)]^{\top}$ と、2 つの定数ベクトル $\mathbf{v}_1 = [a_1, b_1, c_1]^{\top}$, $\mathbf{v}_2 = [a_2, b_2, c_2]^{\top}$ があるとす。このとき以下に示すような、外積のノルムの二乗を考える。

$$\begin{aligned}
& \|(\mathbf{v}(x) - \mathbf{v}_1) \times (\mathbf{v}(x) - \mathbf{v}_2)\|^2 \\
&= \|\mathbf{v}(x) \times \mathbf{v}(x) - \mathbf{v}(x) \times \mathbf{v}_2 - \mathbf{v}_1 \times \mathbf{v}(x) + \mathbf{v}_1 \times \mathbf{v}_2\|^2 \\
&= \|-\mathbf{v}(x) \times \mathbf{v}_2 + \mathbf{v}(x) \times \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2\|^2 \\
&= \|\mathbf{v}(x) \times (\mathbf{v}_1 - \mathbf{v}_2) + \mathbf{v}_1 \times \mathbf{v}_2\|^2 \\
&= \left\| [f(x), g(x), h(x)]^{\top} \times [a_1 - a_2, b_1 - b_2, c_1 - c_2]^{\top} + [a_1, b_1, c_1]^{\top} \times [a_2, b_2, c_2]^{\top} \right\|^2 \\
&= \|(g(x)(c_1 - c_2) - h(x)(b_1 - b_2) + b_1c_2 - c_1b_2, \\
&\quad h(x)(a_1 - a_2) - f(x)(c_1 - c_2) + c_1a_2 - a_1c_2, \\
&\quad f(x)(b_1 - b_2) - g(x)(a_1 - a_2) + a_1b_2 - b_1a_2)\|^2 \\
&= (g(x)(c_1 - c_2) - h(x)(b_1 - b_2) + b_1c_2 - c_1b_2)^2 + \\
&\quad (h(x)(a_1 - a_2) - f(x)(c_1 - c_2) + c_1a_2 - a_1c_2)^2 + \\
&\quad (f(x)(b_1 - b_2) - g(x)(a_1 - a_2) + a_1b_2 - b_1a_2)^2
\end{aligned} \tag{38}$$

これを x で微分すると次を得る。

$$\begin{aligned}
& \frac{d}{dx} \|(\mathbf{v}(x) - \mathbf{v}_1) \times (\mathbf{v}(x) - \mathbf{v}_2)\|^2 \\
&= \frac{d}{dx} (g(x)(c_1 - c_2) - h(x)(b_1 - b_2) + b_1c_2 - c_1b_2)^2 + \\
&\quad (h(x)(a_1 - a_2) - f(x)(c_1 - c_2) + c_1a_2 - a_1c_2)^2 + \\
&\quad (f(x)(b_1 - b_2) - g(x)(a_1 - a_2) + a_1b_2 - b_1a_2)^2 \\
&= 2(g(x)(c_1 - c_2) - h(x)(b_1 - b_2) + b_1c_2 - c_1b_2)(g'(x)(c_1 - c_2) - h'(x)(b_1 - b_2)) \\
&\quad + 2(h(x)(a_1 - a_2) - f(x)(c_1 - c_2) + c_1a_2 - a_1c_2)(h'(x)(a_1 - a_2) - f'(x)(c_1 - c_2)) \\
&\quad + 2(f(x)(b_1 - b_2) - g(x)(a_1 - a_2) + a_1b_2 - b_1a_2)(f'(x)(b_1 - b_2) - g'(x)(a_1 - a_2)) \\
&= 2((\mathbf{v}(x) - \mathbf{v}_1) \times (\mathbf{v}(x) - \mathbf{v}_2))^{\top} \left([f'(x), g'(x), h'(x)]^{\top} \times [a_1 - a_2, b_1 - b_2, c_1 - c_2]^{\top} \right) \\
&= 2((\mathbf{v}(x) - \mathbf{v}_1) \times (\mathbf{v}(x) - \mathbf{v}_2)) \left(\frac{d}{dx} \mathbf{v}(x)^{\top} \times (\mathbf{v}_1 - \mathbf{v}_2) \right) \\
&= 2 \frac{d}{dx} \mathbf{v}(x)^{\top} ((\mathbf{v}_1 - \mathbf{v}_2) \times ((\mathbf{v}(x) - \mathbf{v}_1) \times (\mathbf{v}(x) - \mathbf{v}_2)))
\end{aligned} \tag{39}$$

また、ベクトル $\mathbf{v}(y)$ のノルムの微分は次のように書ける。

$$\begin{aligned}
\frac{d}{dy} \|\mathbf{v}(y)\| &= \frac{d}{dy} \left(\|\mathbf{v}(y)\|^2 \right)^{\frac{1}{2}} \\
&= \frac{1}{2} \left(\|\mathbf{v}(y)\|^2 \right)^{-\frac{1}{2}} \frac{d}{dy} \|\mathbf{v}(y)\|^2 \\
&= \frac{1}{2} \|\mathbf{v}(y)\|^{-1} \frac{d}{dy} \|\mathbf{v}(y)\|^2
\end{aligned} \tag{40}$$

上式を使って、 $f_{\mathcal{E}}$ の T_{k+1}^L に関する勾配ベクトル $\nabla f_{\mathcal{E}}$ を計算する。この勾配ベクトル $\nabla f_{\mathcal{E}}$ は、 T_{k+1}^L の各要素 $t_x, t_y, t_z, \theta_x, \theta_y, \theta_z$ に関する、 $f_{\mathcal{E}}$ の偏微分を並べたものである。

$$\nabla f_{\mathcal{E}} = \left[\frac{\partial f_{\mathcal{E}}}{\partial t_x}, \frac{\partial f_{\mathcal{E}}}{\partial t_y}, \frac{\partial f_{\mathcal{E}}}{\partial t_z}, \frac{\partial f_{\mathcal{E}}}{\partial \theta_x}, \frac{\partial f_{\mathcal{E}}}{\partial \theta_y}, \frac{\partial f_{\mathcal{E}}}{\partial \theta_z} \right]^{\top} \quad (41)$$

以下では簡単のため $\mathbf{R} = \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z)$, $\mathbf{t} = [st_x, st_y, st_z]^{\top}$ とおく。 t_x に関する $f_{\mathcal{E}}$ の偏微分は次のようになる。

$$\begin{aligned} & \frac{\partial}{\partial t_x} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \\ &= \frac{\partial}{\partial t_x} \left\| \bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right\|^{-1} \left\| \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,l)}^L \right) \right\| \\ &= \left\| \bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right\|^{-1} \frac{\partial}{\partial t_x} \left\| \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,l)}^L \right) \right\| \\ &= \frac{1}{2} \left\| \bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right\|^{-1} \left\| \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,l)}^L \right) \right\|^{-1} \\ & \quad \frac{\partial}{\partial t_x} \left\| \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,l)}^L \right) \right\|^2 \\ &= \frac{1}{2} \left(\left\| \bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right\| \left\| \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,l)}^L \right) \right\| \right)^{-1} \\ & \quad 2 \left(\frac{\partial}{\partial t_x} \mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) \right)^{\top} \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \right. \\ & \quad \left. \times \left(\left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,l)}^L \right) \right) \right) \\ &= \left(\left\| \bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right\| \left\| \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \right\| \right)^{-1} \left(\frac{\partial}{\partial t_x} \mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) \right)^{\top} \\ & \quad \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,j)}^L \right) \times \left(\widetilde{\mathbf{X}}_{(k+1,i)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \right) \right) \end{aligned} \quad (42)$$

上記の式変形では分かりやすさのため、 $\widetilde{\mathbf{X}}_{(k+1,i)}^L = \mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right)$ を使って式を短縮している。(24) 式の単位法線ベクトル $[l_x, l_y, l_z]^{\top}$ を使えば以下を得る。

$$\begin{aligned} & \frac{\partial}{\partial t_x} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \\ &= \left(\frac{\partial}{\partial t_x} \mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) \right)^{\top} [l_x, l_y, l_z]^{\top} \\ &= \left(\frac{\partial}{\partial t_x} \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z) \left(\mathbf{X}_{(k+1,i)}^L - [st_x, st_y, st_z]^{\top} \right) \right)^{\top} [l_x, l_y, l_z]^{\top} \end{aligned} \quad (43)$$

$t_y, t_z, \theta_x, \theta_y, \theta_z$ に対する偏微分も、上と同様の式になるから、あとは $\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right)$ の各変数に関する偏微分が分かればよい。

$f_{\mathcal{H}}$ の T_{k+1}^L に関する勾配ベクトル $\nabla f_{\mathcal{H}}$ は、 $f_{\mathcal{E}}$ よりは求めやすい。 $f_{\mathcal{E}}$ では変数 T_{k+1}^L が2箇所が登場したが、 $f_{\mathcal{H}}$ では1箇所だけだからである。勾配ベクトル $\nabla f_{\mathcal{H}}$ は、 T_{k+1}^L の各要素 $t_x, t_y, t_z, \theta_x, \theta_y, \theta_z$ に関する、 $f_{\mathcal{H}}$ の偏微分を並べたものである。

$$\nabla f_{\mathcal{H}} = \left[\frac{\partial f_{\mathcal{H}}}{\partial t_x}, \frac{\partial f_{\mathcal{H}}}{\partial t_y}, \frac{\partial f_{\mathcal{H}}}{\partial t_z}, \frac{\partial f_{\mathcal{H}}}{\partial \theta_x}, \frac{\partial f_{\mathcal{H}}}{\partial \theta_y}, \frac{\partial f_{\mathcal{H}}}{\partial \theta_z} \right]^{\top} \quad (44)$$

先程と同様に $\mathbf{R} = \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z)$, $\mathbf{t} = [st_x, st_y, st_z]^\top$ とおく. t_x に関する $f_{\mathcal{H}}$ の偏微分は次のようになる.

$$\begin{aligned}
& \frac{\partial}{\partial t_x} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \\
&= \frac{\partial}{\partial t_x} \left\| \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right\|^{-1} \\
&\quad \left(\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) - \bar{\mathbf{X}}_{(k,j)}^L \right)^\top \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right) \\
&= \left\| \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right\|^{-1} \\
&\quad \left(\frac{\partial}{\partial t_x} \mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) \right)^\top \left(\left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,l)}^L \right) \times \left(\bar{\mathbf{X}}_{(k,j)}^L - \bar{\mathbf{X}}_{(k,m)}^L \right) \right)
\end{aligned}$$

(31) 式の単位法線ベクトル $[p_x/p_s, p_y/p_s, p_z/p_s]^\top$ を使えば以下を得る.

$$\begin{aligned}
& \frac{\partial}{\partial t_x} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \\
&= \left(\frac{\partial}{\partial t_x} \mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) \right)^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \\
&= \left(\frac{\partial}{\partial t_x} \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z) \left(\mathbf{X}_{(k+1,i)}^L - [st_x, st_y, st_z]^\top \right) \right)^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \quad (45)
\end{aligned}$$

$t_y, t_z, \theta_x, \theta_y, \theta_z$ に対する偏微分も, 上と同様の式になるから, あとは $\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right)$ の各変数に対する偏微分が分かればよい. (43) 式と (45) 式は同じ形をしているので, 関数 $f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$, $f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ の偏微分の計算は, 1 つのソースコードに共通化できる.

回転行列 $\mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z)$ の各要素は次のように表される.

$$\begin{aligned}
& \mathbf{R}(-s\theta_x, -s\theta_y, -s\theta_z) \\
&= (\mathbf{R}_y(-s\theta_y) \mathbf{R}_x(-s\theta_x) \mathbf{R}_z(-s\theta_z)) \\
&= \begin{bmatrix} \cos(-s\theta_y) \cos(-s\theta_z) + \sin(-s\theta_y) \sin(-s\theta_x) \sin(-s\theta_z) & & \\ & \cos(-s\theta_x) \sin(-s\theta_z) & \\ -\sin(-s\theta_y) \cos(-s\theta_z) + \cos(-s\theta_y) \sin(-s\theta_x) \sin(-s\theta_z) & & \\ & -\cos(-s\theta_y) \sin(-s\theta_z) + \sin(-s\theta_y) \sin(-s\theta_x) \cos(-s\theta_z) & \sin(-s\theta_y) \cos(-s\theta_x) \\ & \cos(-s\theta_x) \cos(-s\theta_z) & -\sin(-s\theta_x) \\ \sin(-s\theta_y) \sin(-s\theta_z) + \cos(-s\theta_y) \sin(-s\theta_x) \cos(-s\theta_z) & \cos(-s\theta_y) \cos(-s\theta_x) & \end{bmatrix} \\
&= \begin{bmatrix} \cos(s\theta_y) \cos(s\theta_z) - \sin(s\theta_y) \sin(s\theta_x) \sin(s\theta_z) & & \\ & -\cos(s\theta_x) \sin(s\theta_z) & \\ \sin(s\theta_y) \cos(s\theta_z) + \cos(s\theta_y) \sin(s\theta_x) \sin(s\theta_z) & & \\ & \cos(s\theta_y) \sin(s\theta_z) + \sin(s\theta_y) \sin(s\theta_x) \cos(s\theta_z) & -\sin(s\theta_y) \cos(s\theta_x) \\ & \cos(s\theta_x) \cos(s\theta_z) & \sin(s\theta_x) \\ \sin(s\theta_y) \sin(s\theta_z) - \cos(s\theta_y) \sin(s\theta_x) \cos(s\theta_z) & \cos(s\theta_y) \cos(s\theta_x) & \end{bmatrix} \quad (46)
\end{aligned}$$

$\mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right)$ の各変数 t_x, t_y, t_z に関する偏微分は次の通りである.

$$\begin{aligned}
\frac{\partial}{\partial t_x} \mathbf{R} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t} \right) &= -\frac{\partial}{\partial t_x} \mathbf{R} \mathbf{t} \\
&= -\frac{\partial}{\partial t_x} [(\cos(s\theta_y) \cos(s\theta_z) - \sin(s\theta_y) \sin(s\theta_x) \sin(s\theta_z)) st_x + \text{Const.},
\end{aligned}$$

$$\begin{aligned}
& (-\cos(s\theta_x)\sin(s\theta_z))st_x + \text{Const.}, \\
& (\sin(s\theta_y)\cos(s\theta_z) + \cos(s\theta_y)\sin(s\theta_x)\sin(s\theta_z))st_x + \text{Const.}]^\top \\
& = -s\mathbf{R}_{:,0} \tag{47}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial t_y}\mathbf{R}\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) &= -\frac{\partial}{\partial t_y}\mathbf{R}\mathbf{t} \\
&= -\frac{\partial}{\partial t_y}[(\cos(s\theta_y)\sin(s\theta_z) + \sin(s\theta_y)\sin(s\theta_x)\cos(s\theta_z))st_y + \text{Const.}, \\
&\quad (\cos(s\theta_x)\cos(s\theta_z))st_y + \text{Const.}, \\
&\quad (\sin(s\theta_y)\sin(s\theta_z) - \cos(s\theta_y)\sin(s\theta_x)\cos(s\theta_z))st_y + \text{Const.}]^\top \\
&= -s\mathbf{R}_{:,1} \tag{48}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial t_z}\mathbf{R}\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) &= -\frac{\partial}{\partial t_z}\mathbf{R}\mathbf{t} \\
&= -\frac{\partial}{\partial t_z}[(-\sin(s\theta_y)\cos(s\theta_x))st_z + \text{Const.}, \\
&\quad (\sin(s\theta_x))st_z + \text{Const.}, \\
&\quad (\cos(s\theta_y)\cos(s\theta_x))st_z + \text{Const.}]^\top \\
&= -s\mathbf{R}_{:,2} \tag{49}
\end{aligned}$$

但し $\mathbf{R}_{:,0}, \mathbf{R}_{:,1}, \mathbf{R}_{:,2}$ はそれぞれ、回転行列 \mathbf{R} の 0, 1, 2 列目を取り出した列ベクトルである。 $\theta_x, \theta_y, \theta_z$ に関する偏微分は次の通りである。

$$\begin{aligned}
& \frac{\partial}{\partial \theta_x}\mathbf{R}\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) \\
&= \left(\frac{\partial}{\partial \theta_x}\mathbf{R}\right)\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) \\
&= s \begin{bmatrix} -\sin(s\theta_y)\cos(s\theta_x)\sin(s\theta_z) & & & \\ \sin(s\theta_x)\sin(s\theta_z) & & & \\ \cos(s\theta_y)\cos(s\theta_x)\sin(s\theta_z) & & & \\ \sin(s\theta_y)\cos(s\theta_x)\cos(s\theta_z) & \sin(s\theta_y)\sin(s\theta_x) & & \\ -\sin(s\theta_x)\cos(s\theta_z) & s\cos(s\theta_x) & & \\ -\cos(s\theta_y)\cos(s\theta_x)\cos(s\theta_z) & -\cos(s\theta_y)\sin(s\theta_x) & & \end{bmatrix} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) \tag{50}
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial}{\partial \theta_y}\mathbf{R}\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) \\
&= \left(\frac{\partial}{\partial \theta_y}\mathbf{R}\right)\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) \\
&= s \begin{bmatrix} -\sin(s\theta_y)\cos(s\theta_z) - \cos(s\theta_y)\sin(s\theta_x)\sin(s\theta_z) & & & \\ 0 & & & \\ \cos(s\theta_y)\cos(s\theta_z) - \sin(s\theta_y)\sin(s\theta_x)\sin(s\theta_z) & & & \\ -\sin(s\theta_y)\sin(s\theta_z) + \cos(s\theta_y)\sin(s\theta_x)\cos(s\theta_z) & -\cos(s\theta_y)\cos(s\theta_x) & & \\ 0 & 0 & & \\ \cos(s\theta_y)\sin(s\theta_z) + \sin(s\theta_y)\sin(s\theta_x)\cos(s\theta_z) & -\sin(s\theta_y)\cos(s\theta_x) & & \end{bmatrix} \left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) \tag{51}
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial}{\partial \theta_z}\mathbf{R}\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right) \\
&= \left(\frac{\partial}{\partial \theta_z}\mathbf{R}\right)\left(\mathbf{X}_{(k+1,i)}^L - \mathbf{t}\right)
\end{aligned}$$

$$= s \begin{bmatrix} -\cos(s\theta_y) \sin(s\theta_z) - \sin(s\theta_y) \sin(s\theta_x) \cos(s\theta_z) \\ -\cos(s\theta_x) \cos(s\theta_z) \\ -\sin(s\theta_y) \sin(s\theta_z) + \cos(s\theta_y) \sin(s\theta_x) \cos(s\theta_z) \\ \cos(s\theta_y) \cos(s\theta_z) - \sin(s\theta_y) \sin(s\theta_x) \sin(s\theta_z) & 0 \\ -\cos(s\theta_x) \sin(s\theta_z) & 0 \\ \sin(s\theta_y) \cos(s\theta_z) + \cos(s\theta_y) \sin(s\theta_x) \sin(s\theta_z) & 0 \end{bmatrix} (\mathbf{X}_{(k+1,i)}^L - \mathbf{t}) \quad (52)$$

これらを先ほどの (43) 式および (45) 式に代入すれば、関数 $f_{\mathcal{E}}$ あるいは $f_{\mathcal{H}}$ の、 \mathbf{T}_{k+1}^L の各成分に関する偏微分が得られる。

$$\frac{\partial}{\partial t_x} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = -s (\mathbf{R}_{:,0})^\top [l_x, l_y, l_z]^\top \quad (53)$$

$$\frac{\partial}{\partial t_y} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = -s (\mathbf{R}_{:,1})^\top [l_x, l_y, l_z]^\top \quad (54)$$

$$\frac{\partial}{\partial t_z} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = -s (\mathbf{R}_{:,2})^\top [l_x, l_y, l_z]^\top \quad (55)$$

$$\frac{\partial}{\partial \theta_x} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_x} \mathbf{R} \right) (\mathbf{X}_{(k+1,i)}^L - \mathbf{t}) \right)^\top [l_x, l_y, l_z]^\top \quad (56)$$

$$\frac{\partial}{\partial \theta_y} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_y} \mathbf{R} \right) (\mathbf{X}_{(k+1,i)}^L - \mathbf{t}) \right)^\top [l_x, l_y, l_z]^\top \quad (57)$$

$$\frac{\partial}{\partial \theta_z} f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_z} \mathbf{R} \right) (\mathbf{X}_{(k+1,i)}^L - \mathbf{t}) \right)^\top [l_x, l_y, l_z]^\top \quad (58)$$

$$\frac{\partial}{\partial t_x} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = -s (\mathbf{R}_{:,0})^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \quad (59)$$

$$\frac{\partial}{\partial t_y} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = -s (\mathbf{R}_{:,1})^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \quad (60)$$

$$\frac{\partial}{\partial t_z} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = -s (\mathbf{R}_{:,2})^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \quad (61)$$

$$\frac{\partial}{\partial \theta_x} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_x} \mathbf{R} \right) (\mathbf{X}_{(k+1,i)}^L - \mathbf{t}) \right)^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \quad (62)$$

$$\frac{\partial}{\partial \theta_y} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_y} \mathbf{R} \right) (\mathbf{X}_{(k+1,i)}^L - \mathbf{t}) \right)^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \quad (63)$$

$$\frac{\partial}{\partial \theta_z} f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_z} \mathbf{R} \right) (\mathbf{X}_{(k+1,i)}^L - \mathbf{t}) \right)^\top \frac{1}{p_s} [p_x, p_y, p_z]^\top \quad (64)$$

ソースコード内の変数とは次のように対応する。

$$\begin{aligned} \text{srx} &= \sin(s\theta_x), & \text{sry} &= \sin(s\theta_y), & \text{srz} &= \sin(s\theta_z) \\ \text{crx} &= \cos(s\theta_x), & \text{cry} &= \cos(s\theta_y), & \text{crz} &= \cos(s\theta_z) \\ \text{tx} &= st_x, & \text{ty} &= st_y, & \text{tz} &= st_z \end{aligned}$$

$$\begin{aligned} \text{arx} &= \frac{\partial f_{\mathcal{E}}}{\partial \theta_x} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial \theta_x}, & \text{ary} &= \frac{\partial f_{\mathcal{E}}}{\partial \theta_y} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial \theta_y}, & \text{arz} &= \frac{\partial f_{\mathcal{E}}}{\partial \theta_z} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial \theta_z} \\ \text{atx} &= \frac{\partial f_{\mathcal{E}}}{\partial t_x} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial t_x}, & \text{aty} &= \frac{\partial f_{\mathcal{E}}}{\partial t_y} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial t_y}, & \text{atz} &= \frac{\partial f_{\mathcal{E}}}{\partial t_z} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial t_z} \end{aligned}$$

変数 s は $s = \frac{t_i - t_{k+1}}{\Delta t}$ に対応する。点群 \mathcal{P}_{k+1} のスキャン開始時刻 t_{k+1} から、点 $i \in \mathcal{P}_{k+1}$ が取得されるまでの時間差 $t_i - t_{k+1}$ を、点群の取得間隔 $\Delta t = t_{k+2} - t_{k+1}$ で割ったものであり、点ごとに異なる値をも

つ (点群 \mathcal{P}_{k+1} 内の最初の点ならば 0, 最後の点ならば 1). しかしソースコードをみると, $s = 1$ として固定されているようである. 係数ベクトル (エッジ, あるいは平面に対する単位法線ベクトル) $[l_x, l_y, l_z]^\top$ および $[p_x/p_s, p_y/p_s, p_z/p_s]^\top$ は既に計算してあって, `_coeffSel` から取り出せる. 点 i ($i \in \mathcal{E}_{k+1}$ あるいは $i \in \mathcal{H}_{k+1}$) の座標 $\mathbf{X}_{(k+1,i)}^L$ も, `_laserCloudOri` から得られる. 従って, 偏微分の計算に必要な要素は全て揃っている.

さて, 距離 $d_{\mathcal{E}} = f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ と $d_{\mathcal{H}} = f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ を \mathbf{T}_{k+1}^L について最小化したい. ガウス・ニュートン法では, 以下に示す距離 (残差) の二乗和が, 最小化の目的関数となる.

$$\mathcal{L}(\mathbf{T}_{k+1}^L) = \sum_{i \in \mathcal{E}_{k+1}} \left(f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \right)^2 + \sum_{i \in \mathcal{H}_{k+1}} \left(f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \right)^2 \quad (65)$$

$$(\mathbf{T}_{k+1}^L)^* = \arg \min_{\mathbf{T}_{k+1}^L} \mathcal{L}(\mathbf{T}_{k+1}^L) \quad (66)$$

残差 $f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$, $f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ は \mathbf{T}_{k+1}^L の関数である. \mathbf{T}_{k+1}^L を, 初期値 $\check{\mathbf{T}}_{k+1}^L$ と, 初期値からの変動 $\Delta \mathbf{T}_{k+1}^L$ の和で表し, 変動 $\Delta \mathbf{T}_{k+1}^L$ の関数として書き直す.

$$f_{\mathcal{E}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L + \Delta \mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad (67)$$

$$f_{\mathcal{H}}(\mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) = f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L + \Delta \mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad (68)$$

この新しい関数を使えば, 目的関数も, 変動についての関数 $\mathcal{L}(\check{\mathbf{T}}_{k+1}^L + \Delta \mathbf{T}_{k+1}^L)$ になるので, これを $\Delta \mathbf{T}_{k+1}^L$ について最小化する. 最適な変動 $(\Delta \mathbf{T}_{k+1}^L)^*$ が得られたら, 初期値を $\check{\mathbf{T}}_{k+1}^L$ から $\check{\mathbf{T}}_{k+1}^L + (\Delta \mathbf{T}_{k+1}^L)^*$ に更新し, 同様の処理を繰り返す. 解が収束するまで, 初期値に対する変動を求めて, 現在の解を更新していく. このようにガウス・ニュートン法は, 残差の二乗和で表される目的関数を, 逐次的に最適化するための手法である.

ガウス・ニュートン法では, 残差 $f_{\mathcal{E}}, f_{\mathcal{H}}$ を 1 次の項までテイラー展開し, 初期値 $\check{\mathbf{T}}_{k+1}^L$ のまわりで線形近似する. ここで, 先ほど苦労して求めた勾配ベクトル $\nabla f_{\mathcal{E}}, \nabla f_{\mathcal{H}}$ が登場する.

$$f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L + \Delta \mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \simeq f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^L \quad (69)$$

$$f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L + \Delta \mathbf{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \simeq f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^L \quad (70)$$

これを目的関数 \mathcal{L} に代入すれば, 以下ようになる.

$$\begin{aligned} \mathcal{L}(\Delta \mathbf{T}_{k+1}^L) &= \sum_{i \in \mathcal{E}_{k+1}} \left(f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^L \right)^2 \\ &\quad + \sum_{i \in \mathcal{H}_{k+1}} \left(f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^L \right)^2 \end{aligned} \quad (71)$$

$\Delta \mathbf{T}_{k+1}^L$ に関する偏微分は, 以下のようになる.

$$\begin{aligned} &\frac{\partial}{\partial \Delta \mathbf{T}_{k+1}^L} \mathcal{L}(\Delta \mathbf{T}_{k+1}^L) \\ &= 2 \sum_{i \in \mathcal{E}_{k+1}} \left(f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^L \right) \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \\ &\quad + 2 \sum_{i \in \mathcal{H}_{k+1}} \left(f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^L \right) \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \end{aligned} \quad (72)$$

ここで, ベクトルの微分に関する以下の関係を用いた.

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{a}^\top \mathbf{x} = \mathbf{a} \quad (73)$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = (\mathbf{A} + \mathbf{A}^\top) \mathbf{x} \quad (74)$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{a} \mathbf{a}^\top \mathbf{x} = (\mathbf{a} \mathbf{a}^\top + (\mathbf{a} \mathbf{a}^\top)^\top) \mathbf{x} = 2 \mathbf{a} \mathbf{a}^\top \mathbf{x} \quad (75)$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} (a + \mathbf{b}^\top \mathbf{x})^2 &= \frac{\partial}{\partial \mathbf{x}} (a + \mathbf{b}^\top \mathbf{x})^\top (a + \mathbf{b}^\top \mathbf{x}) \\ &= \frac{\partial}{\partial \mathbf{x}} (a^2 + 2a\mathbf{b}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{b} \mathbf{b}^\top \mathbf{x}) \\ &= 2a\mathbf{b} + 2\mathbf{b} \mathbf{b}^\top \mathbf{x} \\ &= 2a\mathbf{b} + 2(\mathbf{b}^\top \mathbf{x}) \mathbf{b} \\ &= 2(a + \mathbf{b}^\top \mathbf{x}) \mathbf{b} \end{aligned} \quad (76)$$

ΔT_{k+1}^L に関する偏微分を 0 とおいて ΔT_{k+1}^L について解けば、 $\mathcal{L}(\Delta T_{k+1}^L)$ を最小化する最適な更新量 $(\Delta T_{k+1}^L)^*$ が得られる。

$$\begin{aligned} \mathbf{0} &= \sum_{i \in \mathcal{E}_{k+1}} \left(f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta T_{k+1}^L \right) \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \\ &\quad + \sum_{i \in \mathcal{H}_{k+1}} \left(f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \Delta T_{k+1}^L \right) \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \end{aligned} \quad (77)$$

これを ΔT_{k+1}^L について解くと

$$\begin{aligned} &\left(\sum_{i \in \mathcal{E}_{k+1}} \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \right. \\ &\quad \left. + \sum_{i \in \mathcal{H}_{k+1}} \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)^\top \right) \Delta T_{k+1}^L \\ &= - \left(\sum_{i \in \mathcal{E}_{k+1}} \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \right. \\ &\quad \left. + \sum_{i \in \mathcal{H}_{k+1}} \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \right) \end{aligned} \quad (78)$$

行列 $\mathbf{J}_{\mathcal{E}}, \mathbf{J}_{\mathcal{H}}$, ベクトル $\mathbf{d}_{\mathcal{E}}, \mathbf{d}_{\mathcal{H}}$ を以下のように定めれば ($i \in \mathcal{E}_{k+1}$ または $i \in \mathcal{H}_{k+1}$)

$$\mathbf{J}_{\mathcal{E}} = \left[\nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,0)}^L) \quad \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (79)$$

$$\mathbf{J}_{\mathcal{H}} = \left[\nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,0)}^L) \quad \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (80)$$

$$\mathbf{d}_{\mathcal{E}} = \left[f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,0)}^L) \quad f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (81)$$

$$\mathbf{d}_{\mathcal{H}} = \left[f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,0)}^L) \quad f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (82)$$

更に、行列 \mathbf{J} とベクトル \mathbf{d} を次のように定めれば

$$\mathbf{J} = \left[\mathbf{J}_{\mathcal{E}}^\top \quad \mathbf{J}_{\mathcal{H}}^\top \right]^\top = \left[\cdots \quad \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \quad \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (83)$$

$$\mathbf{d} = \left[\mathbf{d}_{\mathcal{E}}^\top \quad \mathbf{d}_{\mathcal{H}}^\top \right]^\top = \left[\cdots \quad f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \quad f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (84)$$

最適な ΔT_{k+1}^L は次の式から得られる.

$$(\mathbf{J}^\top \mathbf{J}) \Delta T_{k+1}^L = -\mathbf{J}^\top \mathbf{d} \quad (85)$$

$$\Rightarrow (\Delta T_{k+1}^L)^* = -(\mathbf{J}^\top \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{d} \quad (86)$$

目的関数 $\mathcal{L}(\check{T}_{k+1}^L + \Delta T_{k+1}^L)$ に, 初期値からの変化 ΔT_{k+1}^L が大きくなり過ぎないための罰則項 (正則化項) を加える. $\text{diag}(\mathbf{J}^\top \mathbf{J})$ は, 6×6 行列 $\mathbf{J}^\top \mathbf{J}$ の対角成分からなる, 6×6 の対角行列である (対角成分以外は全て 0). また λ は正則化の強さを表すパラメータであり, λ を大きくすると, 初期値からの変化 ΔT_{k+1}^L がより小さくなるように最適化される.

$$\lambda \text{diag}(\mathbf{J}^\top \mathbf{J}) \|\Delta T_{k+1}^L\|^2 = \lambda \text{diag}(\mathbf{J}^\top \mathbf{J}) (\Delta T_{k+1}^L)^\top \Delta T_{k+1}^L \quad (87)$$

この場合, 目的関数の偏微分に足される項は

$$\frac{\partial}{\partial \Delta T_{k+1}^L} \lambda \text{diag}(\mathbf{J}^\top \mathbf{J}) (\Delta T_{k+1}^L)^\top \Delta T_{k+1}^L = 2\lambda \text{diag}(\mathbf{J}^\top \mathbf{J}) \Delta T_{k+1}^L \quad (88)$$

であるから, 最適な ΔT_{k+1}^L は次のようになる (論文の (12)).

$$(\Delta T_{k+1}^L)^* = -(\mathbf{J}^\top \mathbf{J} + \lambda \text{diag}(\mathbf{J}^\top \mathbf{J}))^{-1} \mathbf{J}^\top \mathbf{d} \quad (89)$$

ソースコード上では, この正則化は行われておらず, (86) 式を使って初期値からの変化 ΔT_{k+1}^L を求めている. 行列とベクトルのサイズを整理しておく, 行列 $\mathbf{J}_\mathcal{E}$ は $|\mathcal{E}_{k+1}| \times 6$, $\mathbf{J}_\mathcal{H}$ は $|\mathcal{H}_{k+1}| \times 6$, \mathbf{J} は $(|\mathcal{E}_{k+1}| + |\mathcal{H}_{k+1}|) \times 6$, ベクトル $\mathbf{d}_\mathcal{E}$ は $|\mathcal{E}_{k+1}|$, $\mathbf{d}_\mathcal{H}$ は $|\mathcal{H}_{k+1}|$, \mathbf{d} は $|\mathcal{E}_{k+1}| + |\mathcal{H}_{k+1}|$ である. 但し $|\mathcal{E}_{k+1}|$ と $|\mathcal{H}_{k+1}|$ は, 点群 \mathcal{E}_{k+1} と \mathcal{H}_{k+1} に含まれる点の個数を表す (最新の点群 \mathcal{P}_{k+1} のなかで, 顕著なコーナーおよび平面に分類された点の個数).

ヤコビ行列 \mathbf{J} と残差ベクトル \mathbf{d} は, ソースコード内の変数 `matA`, `matB` に対応する. 各点 $i \in \mathcal{E}_{k+1}$, $i \in \mathcal{H}_{k+1}$ に対して, 現在の姿勢 \check{T}_{k+1}^L (`_transform`) のもとで勾配 $\nabla f_\mathcal{E}(\check{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$, $\nabla f_\mathcal{H}(\check{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ を計算する. この勾配 (変数 `arx`, `ary`, `arz`, `atx`, `aty`, `atz`) を, \mathbf{J} の点 i に対応する行 (`matA(i, 0)` から `matA(i, 5)`) に格納する. また, 点 i に対応する距離 (残差) $d_\mathcal{E} = f_\mathcal{E}(\check{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$, $d_\mathcal{H} = f_\mathcal{H}(\check{T}_{k+1}^L, \mathbf{X}_{(k+1,i)}^L)$ も既に計算されているが, これを \mathbf{d} の点 i に対応する要素 `matB(i, 0)` に格納する. 但し, 距離の符号は反転されているので, `matB` が表すベクトルは $-\mathbf{d}$ と対応する. `matAt` は \mathbf{J}^\top , `matAtA` は $\mathbf{J}^\top \mathbf{J}$, `matAtB` は $-\mathbf{J}^\top \mathbf{d}$ に対応する. (86) 式の結果, 即ち $\mathbf{J}^\top \mathbf{J} \Delta T_{k+1}^L = -\mathbf{J}^\top \mathbf{d}$ を満たす最適な更新量 ΔT_{k+1}^L が `matX` に代入される.

6×6 の逆行列 $(\mathbf{J}^\top \mathbf{J})^{-1}$ は, 求めた最適な更新量 ΔT_{k+1}^L の共分散として捉えられる. $(\mathbf{J}^\top \mathbf{J})^{-1}$ の固有値と固有ベクトルは, ΔT_{k+1}^L の分散がどの方向 (固有ベクトル) にどのくらい (固有値) 広がっているのかを表す. 大きな固有値をもつ固有ベクトルがあるとき, そのベクトルの方向に関して ΔT_{k+1}^L は不正確といえる. 例えば LiDAR が長い廊下をスキャンする場合を考えると, スキャンされた点群の形状は, 廊下の壁を沿う 2 本の直線となる. 2 つの点群を重ね合わせると, 廊下に垂直な方向については正確な位置が求まるが, 廊下の長手方向については曖昧さが残ってしまう. この状態は縮退 (Degeneration) とよばれる. 逆行列 $(\mathbf{J}^\top \mathbf{J})^{-1}$ の固有値は, 元の行列 $\mathbf{J}^\top \mathbf{J}$ の固有値の逆数であるから, $\mathbf{J}^\top \mathbf{J}$ において小さな固有値に対応する固有ベクトルの方向に, ΔT_{k+1}^L は不正確である. ΔT_{k+1}^L を使って姿勢の解 T_{k+1}^L を更新する際, この不正確な方向に姿勢を更新してしまわないように ΔT_{k+1}^L を修正したい. 文献 [2] の手法に倣って, 次のように修正できる.

6×6 行列 $\mathbf{J}^\top \mathbf{J}$ の固有値を $\lambda_1, \dots, \lambda_6$ ($\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_6$), 固有ベクトルを $\mathbf{v}_1, \dots, \mathbf{v}_6$ として, 固有ベクトルを並べた行列を $\mathbf{V}_f = [\mathbf{v}_1, \dots, \mathbf{v}_6]^\top$ とおく. $\lambda_1, \dots, \lambda_m$ までの固有値が閾値 (`eignThre`) を下回ってお

り, ΔT_{k+1}^L が固有ベクトル v_1, \dots, v_m の方向に縮退していたとする. 行列 V_f のうち, 最初の m 列を 0 で置き換えてできた新たな行列を $V_u = [0, \dots, 0, v_{m+1}, \dots, v_6]^\top$ とおくと, ΔT_{k+1}^L は次のように修正される.

$$\Delta T_{k+1}^L \leftarrow V_f^{-1} V_u \Delta T_{k+1}^L \quad (90)$$

V_f^\top は matV , V_u^\top は matV2 に対応するから, 姿勢の解 T_{k+1}^L を表す変数 matX に, $\text{matV2.transpose}()$ と $\text{matV.transpose}().\text{inverse}()$ を適用すればよい (線形代数ライブラリとして Eigen を利用する場合). なお, matP は $V_f^{-1} V_u$ に対応し, これを matX に適用している. 実際のソースコードでは, matP として $\text{matV.inverse}()$ と matV2 の積, 即ち $(V_f^\top)^{-1} V_u^\top = (V_f^{-1})^\top V_u^\top = (V_u V_f^{-1})^\top$ を計算しているが, これは誤りと考えられる. 本来は $\text{matV2.transpose}()$ と $\text{matV.transpose}().\text{inverse}()$ の積を matP に格納する必要がある.

以上で, ガウス・ニュートン法の 1 回のイテレーションにおける処理が終わって, 初期値 $\text{transform}(T_{k+1}^L)$ からの更新量 $\text{matX}(\Delta T_{k+1}^L)$ が得られた. transform に matX の各要素を足し合わせて ($T_{k+1}^L \leftarrow T_{k+1}^L + \Delta T_{k+1}^L$), 現在の姿勢を更新する. 姿勢の各要素が NaN や無限大などの不正な値になった場合は, 強制的に 0 にリセットする. 更新量 ΔT_{k+1}^L の並進成分と回転成分のノルム deltaT , deltaR を計算し, 双方が閾値 deltaTAbort , deltaRAbort 未満であれば, 解は収束したものとみなして, ガウス・ニュートン法による更新を終了し, 現在の T_{k+1}^L を最終的な解として返す. そうでなければ, 新しい解のもとで対応付けと距離計算を行って, 再度解を更新する (解 T_{k+1}^L の更新に伴って, 時刻 t_{k+1} における点の座標 $\widetilde{X}_{(k+1,i)}^L$ が変化するため, 点群間の対応付けからやり直す必要がある).

$\text{transform}(T_{k+1}^L)$ は, 時刻 t_{k+1} から時刻 t_{k+2} までの姿勢の変化を表す. transform と, 時刻 t_{k+1} における LiDAR の姿勢 transformSum を合わせることで, 時刻 t_{k+2} における姿勢が得られる. もう少し具体的に記述すると, $T_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$ は以下の式により, 点の座標を, 時刻 t_{k+1} の LiDAR 座標系から時刻 t_{k+2} の LiDAR 座標系に変換する. 時刻 t_k の座標系で表される点の座標を p_k とすれば, この変換は次のように書ける.

$$\begin{aligned} p_{k+2} &= \mathbf{R}(-\theta_x, -\theta_y, -\theta_z)^\top p_{k+1} - [-t_x, -t_y, -t_z]^\top \\ &= (\mathbf{R}_y(-\theta_y) \mathbf{R}_x(-\theta_x) \mathbf{R}_z(-\theta_z))^\top p_{k+1} + [t_x, t_y, t_z]^\top \end{aligned}$$

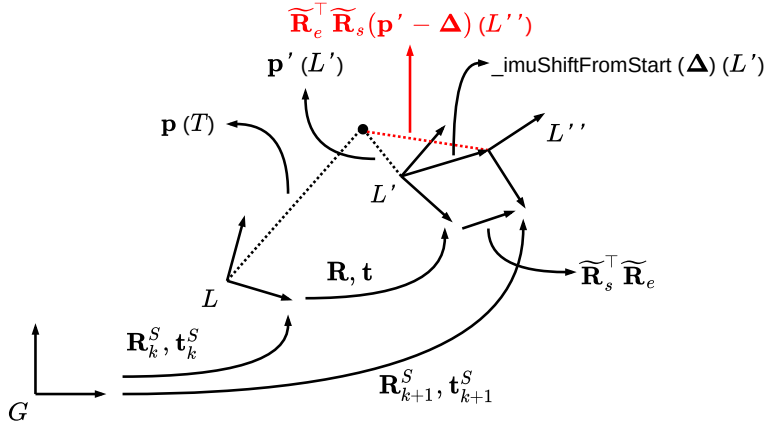
transformSum は, グローバル座標系 (最初の時刻 t_1 における LiDAR 座標系) からみた, 時刻 t_{k+1} における LiDAR 座標系の姿勢であるから, 点の座標を, 時刻 t_{k+1} の LiDAR 座標系からグローバル座標系へと移す. 時刻 t_{k+1} における transformSum を $T_k^S = [t_x^k, t_y^k, t_z^k, \theta_x^k, \theta_y^k, \theta_z^k]^\top$ (論文の表記に合わせている), グローバル座標系での点の座標を p_G とすると, この変換は次のように書ける.

$$\begin{aligned} p_G &= \mathbf{R}(\theta_x^k, \theta_y^k, \theta_z^k) p_{k+1} + [t_x^k, t_y^k, t_z^k]^\top \\ &= (\mathbf{R}_y(\theta_y^k) \mathbf{R}_x(\theta_x^k) \mathbf{R}_z(\theta_z^k)) p_{k+1} + [t_x^k, t_y^k, t_z^k]^\top \end{aligned} \quad (91)$$

p_{k+1} と p_{k+2} , p_{k+1} と p_G との関係式を組み合わせると, 以下を得る.

$$\begin{aligned} p_G &= \mathbf{R}(\theta_x^k, \theta_y^k, \theta_z^k) \left(\mathbf{R}(-\theta_x, -\theta_y, -\theta_z) \left(p_{k+2} - [t_x, t_y, t_z]^\top \right) \right) + [t_x^k, t_y^k, t_z^k]^\top \\ &= (\mathbf{R}(\theta_x^k, \theta_y^k, \theta_z^k) \mathbf{R}(-\theta_x, -\theta_y, -\theta_z)) p_{k+2} \\ &\quad + \left([t_x^k, t_y^k, t_z^k]^\top - (\mathbf{R}(\theta_x^k, \theta_y^k, \theta_z^k) \mathbf{R}(-\theta_x, -\theta_y, -\theta_z)) [t_x, t_y, t_z]^\top \right) \end{aligned} \quad (92)$$

$$= \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) p_{k+2} + [t'_x, t'_y, t'_z]^\top \quad (93)$$



G : グローバル座標系
 L : t_{k+1} におけるLiDAR座標系
 L' : t_{k+2} におけるLiDAR座標系 (補正後)
 L'' : t_{k+2} におけるLiDAR座標系 (補正前)
 $\mathbf{R} = \mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z)$
 $\mathbf{t} = [-t_x, -t_y, -t_z]^\top$
 $\mathbf{T}_{k+1}^L = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$ (transform)
 $\mathbf{T}_k^S = [t_x^k, t_y^k, t_z^k, \theta_x^k, \theta_y^k, \theta_z^k]^\top$
 $\mathbf{R}_k^S = \mathbf{R}_y(\theta_y^k)\mathbf{R}_x(\theta_x^k)\mathbf{R}_z(\theta_z^k)$
 $\mathbf{t}_k^S = [t_x^k, t_y^k, t_z^k]^\top$

$\widetilde{\mathbf{R}}_s = \mathbf{R}_y(\widetilde{\psi}_s)\mathbf{R}_x(\widetilde{\theta}_s)\mathbf{R}_z(\widetilde{\phi}_s)$
 $\widetilde{\mathbf{R}}_e = \mathbf{R}_y(\widetilde{\psi}_e)\mathbf{R}_x(\widetilde{\theta}_e)\mathbf{R}_z(\widetilde{\phi}_e)$
 $\widetilde{\phi}_s, \widetilde{\theta}_s, \widetilde{\psi}_s$: `_imu{Roll, Pitch, Yaw}Start`
 $\widetilde{\phi}_e, \widetilde{\theta}_e, \widetilde{\psi}_e$: `_imu{Roll, Pitch, Yaw}End`

図 10 時刻 t_{k+1} から t_{k+2} へのオドメトリによる姿勢の更新

上式は、時刻 t_{k+2} の LiDAR 座標系で表される点の座標 p_{k+2} を、グローバル座標系での座標 p_G に変換している。上記の姿勢 $[t'_x, t'_y, t'_z, \theta'_x, \theta'_y, \theta'_z]^\top$ は、グローバル座標系から見た、時刻 t_{k+2} の LiDAR 座標系の姿勢である。ここで何度も記している通り、点の座標 p_{k+2} は実際には IMU データによってある程度補正され、加減速による影響が取り除かれているから、 p_{k+2} は時刻 t_{k+2} の本来の LiDAR 座標系 (図 10 の座標系 L'') ではなく、補正後の座標系で表される (図 10 の座標系 L')。従って姿勢 $[t'_x, t'_y, t'_z, \theta'_x, \theta'_y, \theta'_z]^\top$ は、グローバル座標系から、補正後の時刻 t_{k+2} の LiDAR 座標系への姿勢変化を表す (図 10 の座標系 G から L')。これは IMU を使う場合の話であり、IMU を使わないのであれば p_{k+2} は補正されていないから、時刻 t_{k+2} における本来の LiDAR 座標系 (図 10 の L'') で表されたままである。このとき、姿勢 $[t'_x, t'_y, t'_z, \theta'_x, \theta'_y, \theta'_z]^\top$ は座標系 G から L'' への姿勢変化、即ち $\mathbf{T}_{k+1}^S = [t_x^{k+1}, t_y^{k+1}, t_z^{k+1}, \theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}]^\top$ に等しくなっている。座標系 G から L' への回転成分 (オイラー角) $\theta'_x, \theta'_y, \theta'_z$ は以下の関係を満たす。

$$\begin{aligned}
 \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) &= \mathbf{R}(\theta_x^k, \theta_y^k, \theta_z^k)\mathbf{R}(-\theta_x, -\theta_y, -\theta_z) \\
 \mathbf{R}_y(\theta'_y)\mathbf{R}_x(\theta'_x)\mathbf{R}_z(\theta'_z) &= \mathbf{R}_y(\theta_y^k)\mathbf{R}_x(\theta_x^k)\mathbf{R}_z(\theta_z^k)\mathbf{R}_y(-\theta_y)\mathbf{R}_x(-\theta_x)\mathbf{R}_z(-\theta_z)
 \end{aligned} \tag{94}$$

このような $\theta'_x, \theta'_y, \theta'_z$ は `accumulateRotation()` によって得られる。即ち、`accumulateRotation()` の呼び出し後の時点では、変数 `rx, ry, rz` には $\theta'_x, \theta'_y, \theta'_z$ が格納されている。座標系 G から L' への並進成分 t'_x, t'_y, t'_z

は次のようにして得られる.

$$\begin{aligned}
[t'_x, t'_y, t'_z]^\top &= [t_x^k, t_y^k, t_z^k]^\top - (\mathbf{R}(\theta_x^k, \theta_y^k, \theta_z^k) \mathbf{R}(-\theta_x, -\theta_y, -\theta_z)) [t_x, t_y, t_z]^\top \\
&= [t_x^k, t_y^k, t_z^k]^\top - \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) [t_x, t_y, t_z]^\top \\
&= [t_x^k, t_y^k, t_z^k]^\top - \mathbf{R}_y(\theta'_y) \mathbf{R}_x(\theta'_x) \mathbf{R}_z(\theta'_z) [t_x, t_y, t_z]^\top
\end{aligned} \tag{95}$$

この $[t'_x, t'_y, t'_z, \theta'_x, \theta'_y, \theta'_z]^\top$ は, グローバル座標系 G から, 定速移動の仮定に基づく補正後の座標系 L' への姿勢変化, 即ちロボットが時刻 t_{k+1} から t_{k+2} までの間定速で移動していた場合の姿勢変化である. いま求めたい $\mathbf{T}_{k+1}^S = [t_x^{k+1}, t_y^{k+1}, t_z^{k+1}, \theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}]^\top$ は, 定速移動に加えて, 加減速による非線形な移動も考慮した姿勢変化, 即ちグローバル座標系 G から本来の時刻 t_{k+2} の座標系 L'' への姿勢変化である. 即ち IMU を使う場合は, 定速移動による姿勢変化のみを最適化処理で推定し, IMU によって得られた, 加減速による非線形な姿勢変化を加えることによって, 最終的な姿勢を算出する. IMU を使わない場合, 上記の $[t'_x, t'_y, t'_z, \theta'_x, \theta'_y, \theta'_z]^\top$ は \mathbf{T}_{k+1}^S に等しくなっており, 最適化処理のみで最終的な姿勢が推定される. IMU による以下の補正は IMU データを使う場合に必要となる. IMU データによる点の座標 p_{k+2} の補正は, 図 10 に示す座標系 L' から L'' への変換に相当し, 次のように記述される.

$$p_{k+2} = \left(\left(\mathbf{R}_y(\tilde{\psi}_s) \mathbf{R}_x(\tilde{\theta}_s) \mathbf{R}_z(\tilde{\phi}_s) \right)^\top \mathbf{R}_y(\tilde{\psi}_e) \mathbf{R}_x(\tilde{\theta}_e) \mathbf{R}_z(\tilde{\phi}_e) \right) p'_{k+2} + \Delta \tag{97}$$

p'_{k+2} は補正される前の座標である. オイラー角 $\tilde{\phi}_s, \tilde{\theta}_s, \tilde{\psi}_s$ は `_imuRollStart`, `_imuPitchStart`, `_imuYawStart`, $\tilde{\phi}_e, \tilde{\theta}_e, \tilde{\psi}_e$ は `_imuRollEnd`, `_imuPitchEnd`, `_imuYawEnd`, Δ は `_imuShiftFromStart` に対応し, 図 10 の L' 座標系で表される. これを p_G と p_{k+2} の関係式に代入すれば, 以下が得られる.

$$\begin{aligned}
p_G &= \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) p_{k+2} + [t'_x, t'_y, t'_z]^\top \\
&= \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) \left(\left(\mathbf{R}_y(\tilde{\psi}_s) \mathbf{R}_x(\tilde{\theta}_s) \mathbf{R}_z(\tilde{\phi}_s) \right)^\top \mathbf{R}_y(\tilde{\psi}_e) \mathbf{R}_x(\tilde{\theta}_e) \mathbf{R}_z(\tilde{\phi}_e) \right) p'_{k+2} \\
&\quad + \left(\mathbf{R}(\theta'_x, \theta'_y, \theta'_z) \Delta + [t'_x, t'_y, t'_z]^\top \right) \\
&= \mathbf{R}(\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}) p'_{k+2} + [t_x^{k+1}, t_y^{k+1}, t_z^{k+1}]^\top
\end{aligned} \tag{98}$$

従って, \mathbf{T}_{k+1}^S の回転成分 (オイラー角) $\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}$ は以下の関係を満たす.

$$\mathbf{R}(\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}) = \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) \left(\left(\mathbf{R}_y(\tilde{\psi}_s) \mathbf{R}_x(\tilde{\theta}_s) \mathbf{R}_z(\tilde{\phi}_s) \right)^\top \mathbf{R}_y(\tilde{\psi}_e) \mathbf{R}_x(\tilde{\theta}_e) \mathbf{R}_z(\tilde{\phi}_e) \right) \tag{99}$$

このような $\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}$ は `pluginIMURotation()` によって得られる. 即ち, `pluginIMURotation()` の呼び出し後の時点で, 変数 `rx`, `ry`, `rz` には $\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}$ が格納されている. これらを `_transformSum.rot_x`, `_transformSum.rot_y`, `_transformSum.rot_z` に代入することで, 姿勢の回転成分を更新している. \mathbf{T}_{k+1}^S の並進成分 $t_x^{k+1}, t_y^{k+1}, t_z^{k+1}$ は次のようにして得られる.

$$\begin{aligned}
[t_x^{k+1}, t_y^{k+1}, t_z^{k+1}]^\top &= \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) \Delta + [t'_x, t'_y, t'_z]^\top \\
&= \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) \Delta + \left([t_x^k, t_y^k, t_z^k]^\top - \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) [t_x, t_y, t_z]^\top \right) \\
&= [t_x^k, t_y^k, t_z^k]^\top + \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) \left(\Delta - [t_x, t_y, t_z]^\top \right) \\
&= [t_x^k, t_y^k, t_z^k]^\top - \mathbf{R}(\theta'_x, \theta'_y, \theta'_z) \left([t_x, t_y, t_z]^\top - \Delta \right)
\end{aligned} \tag{100}$$

ソースコード内では上式を計算するため、最初に `_transform.pos` から `_imuShiftFromStart` を引いたベクトル $v([t_x, t_y, t_z]^T - \Delta)$ を用意し、`rotateZXY()` によってオイラー角 `rx`, `ry`, `rz` の分だけ回転させている (この時点で右辺第 2 項が得られる)。このときの `rx`, `ry`, `rz` は `pluginIMURotation()` を呼び出す前のものであり、 $\theta'_x, \theta'_y, \theta'_z$ を表す。続いて、`_transformSum.pos` ($[t_x^k, t_y^k, t_z^k]^T$) から `v` を引いたベクトル `trans` を計算している (T_{k+1}^S の並進成分 $[t_x^{k+1}, t_y^{k+1}, t_z^{k+1}]^T$)。この `trans` を `_transformSum.pos` に代入することで、姿勢の並進成分を更新している。

ここまでの処理で、時刻 t_{k+2} における LiDAR 座標系の姿勢 T_{k+1}^S が求まった。

`_lastCornerCloud` と `_lastSurfaceCloud` は、 k 番目の点群においてコーナー、あるいは平面として分類された点を集めたものである。現在の $k+1$ 番目の点群のうち、コーナーあるいは点群として分類されたもの、即ち `_cornerPointsLessSharp` と `_surfPointsLessFlat` を、`_lastCornerCloud` と `_lastSurfaceCloud` にそれぞれ格納し、 $k+2$ 番目の次の点群に向けて準備しておく。また、`_lastCornerCloud` と `_lastSurfaceCloud` から KD 木 `_lastCornerCloud` と `_lastSurfaceCloud` を作成しておく (最近傍点の探索で利用される)。ScanRegistration ノード内での前処理によって、2 つの点群に含まれる各点の座標は、定速移動の仮定の下で補正されている (図 10 の座標系 L)。 `transformToEnd()` を呼び出すことによって、この補正を取り除いて (加減速による非線形な移動を考慮して)、本来の点の座標に戻している (図 10 の座標系 L'')。

`_lastCornerCloud` と `_lastSurfaceCloud` は、この時点ではまだ現在の ($k+1$ 番目の) 点群を表しており、`/laser_cloud_corner_last` トピックと `/laser_cloud_surf_last` トピックから Publish される。時刻 t_{k+2} における LiDAR の姿勢 T_{k+1}^S は、`/laser_odom_to_init` トピックから Publish される。そして $k+1$ 番目の点群は、`transformToEnd()` を呼んで (加減速による非線形な移動を考慮して) から `/velodyne_cloud_3` トピックを通じて Publish される。結局、LaserOdometry ノードから Publish される点群は、各点が図 10 の座標系 L'' で表現されるほか、座標系 G からみた座標系 L'' の姿勢が Publish される。

IMU データを使う場合は先述の通り、最新の ($k+1$ 番目の) 点群から加減速の影響を取り除いたうえで、直前の点群とのレジストレーションを行って、時刻 t_{k+1} から t_{k+2} にかけての LiDAR の姿勢変化を算出している。計算された姿勢変化は、LiDAR が時刻 t_{k+1} から t_{k+2} までの間定速移動していた場合のものであるから、加減速の影響を再度考慮したうえで Publish する。最新の点群についても、各点の座標に加減速の影響を加えて修正したうえで Publish する。LaserOdometry ノードではこのように、定速移動による姿勢変化と、加減速 (非線形な移動) による姿勢変化の 2 つを切り分けて考えている。定速移動による姿勢変化を点群同士のレジストレーションから計算し、加減速による姿勢変化は IMU データから得て、両者を最後に結合することによって、LiDAR の姿勢変化を求めている。IMU データを使わない場合は、点群同士のレジストレーションのみから姿勢変化を算出する。

6 LaserOdometry クラス

6.1 LaserOdometry::setup()

ノードの設定を ROS から受け取って、次のパラメータを設定する。

- `_scanPeriod(BasicLaserOdometry)`: 点群の取得間隔 (デフォルトでは 100ms)。
- `_ioRatio(LaserOdometry)`: 何回に一回、点群をトピックから Publish するかを決めるための頻度 (デフォルトでは 2)。
- `_maxIterations(BasicLaserOdometry)`: ガウス・ニュートン法の最大のイテレーション数 (デフォ

ルトでは 25).

- `_deltaTAbort(BasicLaserOdometry)`: ガウス・ニュートン法の解 (姿勢) の収束判定に用いる, 並進成分の変化量の閾値 (デフォルトでは 0.1).
- `_deltaRAbort(BasicLaserOdometry)`: ガウス・ニュートン法の解 (姿勢) の収束判定に用いる, 回転成分の変化量の閾値 (デフォルトでは 0.1).

続いて, ノードが Subscribe あるいは Publish するトピックを設定する. Subscribe するのは次の 6 つのトピックである. これらは ScanRegistration が Publish するトピックと同一であるため, 説明を省略する. 5 種類の点群を Subscribe するが, これらは全て, 点群のスキャン開始時刻に揃えられている (MultiScanRegistration の `process()` 内で, BasicScanRegistration の `projectPointToStartOfSweep()` を呼び出すため). 点群を Publish する際は, これらを全て, 点群のスキャン終了時刻 (次のスイープの開始時刻, あるいは次の点群のスキャン開始時刻) に揃える (BasicLaserOdometry の `transformToEnd()` を呼び出し, オドメトリの結果を使う).

- `/laser_cloud_sharp` トピック (sensor_msgs::PointCloud2 型)
- `/laser_cloud_less_sharp` トピック (sensor_msgs::PointCloud2 型)
- `/laser_cloud_flat` トピック (sensor_msgs::PointCloud2 型)
- `/laser_cloud_less_flat` トピック (sensor_msgs::PointCloud2 型)
- `/velodyne_cloud_2` トピック (sensor_msgs::PointCloud2 型)
- `/imu_trans` トピック (sensor_msgs::PointCloud2 型)

Publish するのは次の 4 つのトピックである. `/laser_cloud_corner_last`, `/laser_cloud_surf_last`, `/velodyne_cloud_3` の 3 つのトピックは, `_ioRatio` 回に 1 回 Publish する (デフォルトでは 2 回に 1 回, 即ち 5Hz の頻度).

- `/laser_cloud_corner_last` トピック (sensor_msgs::PointCloud2 型): ScanRegistration の `/laser_cloud_less_sharp` トピックから受け取った $k+1$ 番目の点群は, それほど顕著ではないコーナー, あるいは顕著なコーナーとして分類された点のセットであり, 点の座標はスキャン開始時刻 ($k+1$ 番目のスイープの開始時刻) t_{k+1} に揃えられている. オドメトリの計算結果 (2 つの隣接する点群間の相対姿勢, 即ち時刻 t_{k+1} から t_{k+2} にかけての LiDAR の姿勢変化) を使って, この点の座標をスキャン終了時刻 (即ち $k+2$ 番目のスイープの開始時刻) t_{k+2} に揃えて, `/laser_cloud_corner_last` トピックから Publish する.
- `/laser_cloud_surf_last` トピック (sensor_msgs::PointCloud2 型): ScanRegistration の `/laser_cloud_less_flat` トピックから受け取った $k+1$ 番目の点群は, それほど顕著ではない平面, あるいは顕著な平面として分類された点のセットであり, 点の座標はスキャン開始時刻 t_{k+1} に揃えられている. オドメトリの計算結果を使って, この点の座標を次のスイープの開始時刻 t_{k+2} に揃えて, `/laser_cloud_surf_last` トピックから Publish する.
- `/velodyne_cloud_3` トピック (sensor_msgs::PointCloud2 型): ScanRegistration の `/velodyne_cloud_2` トピックから受け取った $k+1$ 番目の点群 \mathcal{P}_{k+1} は, 座標がスキャン開始時刻 t_{k+1} に揃えられている. オドメトリの計算結果を使って, この点の座標を次のスイープの開始時刻 t_{k+2} に揃えて $\bar{\mathcal{P}}_{k+1}$ を作成し, `/velodyne_cloud_3` トピックから Publish する.
- `/laser_odom_to_init` トピック (nav_msgs::Odometry 型): オドメトリによって得られた, 時刻

t_{k+2} における LiDAR の姿勢 (グローバル座標系). camera_init フレーム (グローバル座標系) からみた/laser_odom フレーム (LiDAR あるいは IMU 座標系) の姿勢.

6.2 LaserOdometry::laserCloudSharpHandler()

/laser_cloud_sharp トピックに紐付けられたコールバックである. 顕著なコーナーとしてラベル付けされた点群を受け取り, 点群のタイムスタンプを_timeCornerPointsSharp にセットする. また ROS の点群メッセージ (sensor_msgs::PointCloud2 型) を, PCL(Point Cloud Library) における点群データ (pcl::PointCloud 型) に変換し, 座標が NaN の点を取り除いたうえで, BasicLaserOdometry の_cornerPointsSharp メンバに格納する.

6.3 LaserOdometry::laserCloudLessSharpHandler()

/laser_cloud_less_sharp トピックに紐付けられたコールバックである. それほど顕著ではないコーナーとしてラベル付けされた点群を受け取り, 点群のタイムスタンプを_timeCornerPointsLessSharp にセットする. また ROS の点群メッセージ (sensor_msgs::PointCloud2 型) を, PCL(Point Cloud Library) における点群データ (pcl::PointCloud 型) に変換し, 座標が NaN の点を取り除いたうえで, BasicLaserOdometry の_cornerPointsLessSharp メンバに格納する.

6.4 LaserOdometry::laserCloudFlatHandler()

/laser_cloud_flat トピックに紐付けられたコールバックである. 顕著な平面としてラベル付けされた点群を受け取り, 点群のタイムスタンプを_timeSurfPointsFlat にセットする. また ROS の点群メッセージ (sensor_msgs::PointCloud2 型) を, PCL(Point Cloud Library) における点群データ (pcl::PointCloud 型) に変換し, 座標が NaN の点を取り除いたうえで, BasicLaserOdometry の_surfPointsFlat メンバに格納する.

6.5 LaserOdometry::laserCloudLessFlatHandler()

/laser_cloud_less_flat トピックに紐付けられたコールバックである. それほど顕著ではない平面としてラベル付けされた点群を受け取り, 点群のタイムスタンプを_timeSurfPointsLessFlat にセットする. また ROS の点群メッセージ (sensor_msgs::PointCloud2 型) を, PCL(Point Cloud Library) における点群データ (pcl::PointCloud 型) に変換し, 座標が NaN の点を取り除いたうえで, BasicLaserOdometry の_surfPointsLessFlat メンバに格納する.

6.6 LaserOdometry::laserCloudFullResHandler()

/velodyne_cloud_2 トピックに紐付けられたコールバックである. 全ての点を含んだ点群を受け取り, 点群のタイムスタンプを_timeLaserCloudFullRes にセットする. また ROS の点群メッセージ (sensor_msgs::PointCloud2 型) を, PCL(Point Cloud Library) における点群データ (pcl::PointCloud 型) に変換し, 座標が NaN の点を取り除いたうえで, BasicLaserOdometry の_laserCloud メンバに格納する.

6.7 LaserOdometry::hasNewData()

ScanRegistration のノードから、新しいデータが届いているかどうかを確認する。Subscribe している 6 つ全てのトピックから新しいデータが届いており、かつそれらのタイムスタンプが揃っている (新しいデータが、同一の点群を表している) 場合に true を返す。

6.8 LaserOdometry::publishResult()

時刻 t_{k+2} における LiDAR の姿勢 (グローバル座標系) を、/laser_odom_to_init トピックから Publish する。この姿勢は、 k 番目の点群 $\bar{\mathcal{P}}_k$ に対する、 $k+1$ 番目の点群 \mathcal{P}_{k+1} のレジストレーションによって得られる。先述の通り、2 つの点群 $\mathcal{P}_{k+1}, \bar{\mathcal{P}}_k$ は、いずれも $k+1$ 番目の点群のスキャン開始時刻 ($k+1$ 番目のスイープの開始時刻) t_{k+1} に揃えられている。

LaserOdometry ノードでは、時間的に隣接する点群同士のレジストレーション (重ね合わせ処理) を繰り返すことで姿勢を算出する (オドメトリという)。後述の LaserMapping ノードでは、点群と地図を使ってレジストレーションを行うことで姿勢を算出する (マッピングという)。オドメトリの計算負荷は低いが、姿勢の各成分には誤差が急速に累積する。マッピングでは、より多くの点を含んだ点群を使用するため計算負荷が高くなるが、そのぶん精度の良い姿勢が得られる。オドメトリによる姿勢は、10Hz の頻度で LaserOdometry ノードから、マッピングによって得られた姿勢は、5Hz の頻度で LaserMapping ノードから Publish される。

時刻 t_{k+2} における LiDAR の姿勢は、transformSum() から取得される (実体は BasicLaserOdometry の _transformSum)。この姿勢の回転成分をオイラー角、即ちロール角 ϕ 、ピッチ角 θ 、ヨー角 ψ で表す。ソースコード内では、 ϕ が transformSum().rot_x、 θ が transformSum().rot_y、 ψ が transformSum().rot_z に対応する。 ϕ, θ, ψ は、LOAM の座標軸における X, Y, Z 軸周りの回転を表しており、LOAM では $Z \rightarrow X \rightarrow Y$ 軸の順に回転させる。姿勢の回転成分を nav_msgs::Odometry 型のメッセージとして Publish するために、(1) Z 軸周りに ψ 、(2) X 軸周りに ϕ 、(3) Y 軸周りに θ という回転、即ち回転行列であれば $R = R_y(\theta)R_x(\phi)R_z(\psi)$ のように記述される回転を、クォータニオン (geometry_msgs::Quaternion 型) を使って表現しなければならない。

ROS の tf::createQuaternionMsgFromRollPitchYaw() を使えば、オイラー角と対応するクォータニオンが得られる。しかし困ったことに、ROS ではオイラー角の回転順序として $\tilde{X} \rightarrow \tilde{Y} \rightarrow \tilde{Z}$ を採用している。この関数に LOAM 内で使っていたオイラー角 ϕ, θ, ψ を与えれば、(1) \tilde{X} 軸周りに ϕ 、(2) \tilde{Y} 軸周りに θ 、(3) \tilde{Z} 軸周りに ψ という回転が得られる。ROS の $\tilde{X}, \tilde{Y}, \tilde{Z}$ 軸は、LOAM では Z, X, Y 軸にそれぞれ対応するので、得られる回転は $\tilde{R}_z(\psi)\tilde{R}_y(\theta)\tilde{R}_x(\phi) = R_y(\psi)R_x(\theta)R_z(\phi)$ となるが、これは求めている回転 $R = R_y(\theta)R_x(\phi)R_z(\psi)$ とは明らかに異なる。従って、 $R_y(\theta)R_x(\phi)R_z(\psi)$ と対応するクォータニオンを計算するには、オイラー角の順序や符号を入れ替えてから tf::createQuaternionMsgFromRollPitchYaw() に渡す必要があり、そのための方法を説明する。

軸 (l_x, l_y, l_z) の周りに角度 Ω だけ回転させるようなクォータニオンは、以下のように表せる。

$$\left(l_x \sin \frac{\Omega}{2}, l_y \sin \frac{\Omega}{2}, l_z \sin \frac{\Omega}{2}, \cos \frac{\Omega}{2} \right) \quad (101)$$

$\tilde{X}, \tilde{Y}, \tilde{Z}$ 軸周りの $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ の回転を表すクォータニオンを $\tilde{\mathbf{q}}_x(\tilde{\phi}), \tilde{\mathbf{q}}_y(\tilde{\theta}), \tilde{\mathbf{q}}_z(\tilde{\psi})$ とおくと、次のように書ける。

$$\tilde{\mathbf{q}}_x(\tilde{\phi}) = \left(\sin \frac{\tilde{\phi}}{2}, 0, 0, \cos \frac{\tilde{\phi}}{2} \right) \quad (102)$$

$$\tilde{\mathbf{q}}_y(\tilde{\theta}) = \left(0, \sin \frac{\tilde{\theta}}{2}, 0, \cos \frac{\tilde{\theta}}{2} \right) \quad (103)$$

$$\tilde{\mathbf{q}}_z(\tilde{\psi}) = \left(0, 0, \sin \frac{\tilde{\psi}}{2}, \cos \frac{\tilde{\psi}}{2} \right) \quad (104)$$

X, Y, Z 軸周りの ϕ, θ, ψ の回転を表すクォータニオン $\mathbf{q}_x(\phi), \mathbf{q}_y(\theta), \mathbf{q}_z(\psi)$ も同様に定められる。クォータニオン $\mathbf{q} = (a_x, a_y, a_z, a_w) = (\mathbf{a}, a_w)$, $\mathbf{q}' = (b_x, b_y, b_z, b_w) = (\mathbf{b}, b_w)$ の積、即ち 2 つの回転の合成は以下のようになる (\times はベクトルの外積)。

$$(a_w \mathbf{b} + b_w \mathbf{a} + \mathbf{a} \times \mathbf{b}, a_w b_w - \mathbf{a}^\top \mathbf{b}) \quad (105)$$

$$= (a_w b_x + b_w a_x + a_y b_z - a_z b_y, a_w b_y + b_w a_y + a_z b_x - a_x b_z, \\ a_w b_z + b_w a_z + a_x b_y - a_y b_x, a_w b_w - a_x b_x - a_y b_y - a_z b_z) \quad (106)$$

さて、 $\tilde{X} \rightarrow \tilde{Y} \rightarrow \tilde{Z}$ の順に $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ だけ回転させるクォータニオン $\tilde{\mathbf{q}}_z(\tilde{\psi})\tilde{\mathbf{q}}_y(\tilde{\theta})\tilde{\mathbf{q}}_x(\tilde{\phi})$ は、次のように計算できる。

$$\begin{aligned} & \tilde{\mathbf{q}}_y(\tilde{\theta})\tilde{\mathbf{q}}_x(\tilde{\phi}) \\ &= \left(0, \sin \frac{\tilde{\theta}}{2}, 0, \cos \frac{\tilde{\theta}}{2} \right) \left(\sin \frac{\tilde{\phi}}{2}, 0, 0, \cos \frac{\tilde{\phi}}{2} \right) \\ &= \left(\cos \frac{\tilde{\theta}}{2} \left(\sin \frac{\tilde{\phi}}{2}, 0, 0 \right) + \cos \frac{\tilde{\phi}}{2} \left(0, \sin \frac{\tilde{\theta}}{2}, 0 \right) + \left(0, \sin \frac{\tilde{\theta}}{2}, 0 \right) \times \left(\sin \frac{\tilde{\phi}}{2}, 0, 0 \right), \right. \\ & \quad \left. \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} - \left(0, \sin \frac{\tilde{\theta}}{2}, 0 \right)^\top \left(\sin \frac{\tilde{\phi}}{2}, 0, 0 \right) \right) \\ &= \left(\cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, -\sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} \right) \end{aligned}$$

であるから

$$\begin{aligned} & \tilde{\mathbf{q}}_z(\tilde{\psi}) \left(\cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, -\sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} \right) \\ &= \left(0, 0, \sin \frac{\tilde{\psi}}{2}, \cos \frac{\tilde{\psi}}{2} \right) \left(\cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, -\sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} \right) \\ &= \left(\cos \frac{\tilde{\psi}}{2} \left(\cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, -\sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2} \right) + \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} \left(0, 0, \sin \frac{\tilde{\psi}}{2} \right) \right. \\ & \quad \left. + \left(0, 0, \sin \frac{\tilde{\psi}}{2} \right) \times \left(\cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, -\sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2} \right), \right. \\ & \quad \left. \cos \frac{\tilde{\psi}}{2} \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} - \left(0, 0, \sin \frac{\tilde{\psi}}{2} \right)^\top \left(\cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, -\sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2} \right) \right) \end{aligned}$$

$$\begin{aligned}
&= \left(\cos \frac{\tilde{\psi}}{2} \cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2} - \sin \frac{\tilde{\psi}}{2} \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, \cos \frac{\tilde{\psi}}{2} \sin \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} + \sin \frac{\tilde{\psi}}{2} \cos \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2}, \right. \\
&\quad \left. - \cos \frac{\tilde{\psi}}{2} \sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2} + \sin \frac{\tilde{\psi}}{2} \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2}, \cos \frac{\tilde{\psi}}{2} \cos \frac{\tilde{\theta}}{2} \cos \frac{\tilde{\phi}}{2} + \sin \frac{\tilde{\psi}}{2} \sin \frac{\tilde{\theta}}{2} \sin \frac{\tilde{\phi}}{2} \right) \quad (107)
\end{aligned}$$

となる. (107) 式の $\tilde{\phi}, \tilde{\theta}, \tilde{\psi}$ を, それぞれ $\psi, -\phi, -\theta$ で置き換えて得られるクォータニオン $\tilde{\mathbf{q}}_z(-\theta)\tilde{\mathbf{q}}_y(-\phi)\tilde{\mathbf{q}}_x(\psi) = (q'_1, q'_2, q'_3, q'_0)$ は以下ようになる.

$$\begin{aligned}
&\tilde{\mathbf{q}}_z(-\theta)\tilde{\mathbf{q}}_y(-\phi)\tilde{\mathbf{q}}_x(\psi) \\
&= \left(\cos \frac{-\theta}{2} \cos \frac{-\phi}{2} \sin \frac{\psi}{2} - \sin \frac{-\theta}{2} \sin \frac{-\phi}{2} \cos \frac{\psi}{2}, \right. \\
&\quad \cos \frac{-\theta}{2} \sin \frac{-\phi}{2} \cos \frac{\psi}{2} + \sin \frac{-\theta}{2} \cos \frac{-\phi}{2} \sin \frac{\psi}{2}, \\
&\quad - \cos \frac{-\theta}{2} \sin \frac{-\phi}{2} \sin \frac{\psi}{2} + \sin \frac{-\theta}{2} \cos \frac{-\phi}{2} \cos \frac{\psi}{2}, \\
&\quad \left. \cos \frac{-\theta}{2} \cos \frac{-\phi}{2} \cos \frac{\psi}{2} + \sin \frac{-\theta}{2} \sin \frac{-\phi}{2} \sin \frac{\psi}{2} \right) \\
&= \left(\cos \frac{\theta}{2} \cos \frac{\phi}{2} \sin \frac{\psi}{2} - \sin \frac{\theta}{2} \sin \frac{\phi}{2} \cos \frac{\psi}{2}, - \cos \frac{\theta}{2} \sin \frac{\phi}{2} \cos \frac{\psi}{2} - \sin \frac{\theta}{2} \cos \frac{\phi}{2} \sin \frac{\psi}{2}, \right. \\
&\quad \left. \cos \frac{\theta}{2} \sin \frac{\phi}{2} \sin \frac{\psi}{2} - \sin \frac{\theta}{2} \cos \frac{\phi}{2} \cos \frac{\psi}{2}, \cos \frac{\theta}{2} \cos \frac{\phi}{2} \cos \frac{\psi}{2} + \sin \frac{\theta}{2} \sin \frac{\phi}{2} \sin \frac{\psi}{2} \right) \quad (108)
\end{aligned}$$

続いて, $Z \rightarrow X \rightarrow Y$ の順に ψ, ϕ, θ だけ回転させるクォータニオン $\mathbf{q}_y(\theta)\mathbf{q}_x(\phi)\mathbf{q}_z(\psi) = (q_1, q_2, q_3, q_0)$ は, 次のように計算できる.

$$\begin{aligned}
&\mathbf{q}_x(\phi)\mathbf{q}_z(\psi) \\
&= \left(\sin \frac{\phi}{2}, 0, 0, \cos \frac{\phi}{2} \right) \left(0, 0, \sin \frac{\psi}{2}, \cos \frac{\psi}{2} \right) \\
&= \left(\cos \frac{\phi}{2} \left(0, 0, \sin \frac{\psi}{2} \right) + \cos \frac{\psi}{2} \left(\sin \frac{\phi}{2}, 0, 0 \right) + \left(\sin \frac{\phi}{2}, 0, 0 \right) \times \left(0, 0, \sin \frac{\psi}{2} \right), \right. \\
&\quad \left. \cos \frac{\phi}{2} \cos \frac{\psi}{2} - \left(\sin \frac{\phi}{2}, 0, 0 \right)^\top \left(0, 0, \sin \frac{\psi}{2} \right) \right) \\
&= \left(\sin \frac{\phi}{2} \cos \frac{\psi}{2}, - \sin \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \cos \frac{\psi}{2} \right)
\end{aligned}$$

であるから

$$\begin{aligned}
&\mathbf{q}_y(\theta) \left(\sin \frac{\phi}{2} \cos \frac{\psi}{2}, - \sin \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \cos \frac{\psi}{2} \right) \\
&= \left(0, \sin \frac{\theta}{2}, 0, \cos \frac{\theta}{2} \right) \left(\sin \frac{\phi}{2} \cos \frac{\psi}{2}, - \sin \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \cos \frac{\psi}{2} \right) \\
&= \left(\cos \frac{\theta}{2} \left(\sin \frac{\phi}{2} \cos \frac{\psi}{2}, - \sin \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \sin \frac{\psi}{2} \right) + \cos \frac{\theta}{2} \cos \frac{\psi}{2} \left(0, \sin \frac{\theta}{2}, 0 \right) \right. \\
&\quad \left. + \left(0, \sin \frac{\theta}{2}, 0 \right) \times \left(\sin \frac{\phi}{2} \cos \frac{\psi}{2}, - \sin \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \sin \frac{\psi}{2} \right), \right. \\
&\quad \left. \cos \frac{\phi}{2} \cos \frac{\psi}{2} \cos \frac{\theta}{2} - \left(0, \sin \frac{\theta}{2}, 0 \right)^\top \left(\sin \frac{\phi}{2} \cos \frac{\psi}{2}, - \sin \frac{\phi}{2} \sin \frac{\psi}{2}, \cos \frac{\phi}{2} \sin \frac{\psi}{2} \right) \right)
\end{aligned}$$

$$= \begin{pmatrix} \cos \frac{\theta}{2} \sin \frac{\phi}{2} \cos \frac{\psi}{2} + \sin \frac{\theta}{2} \cos \frac{\phi}{2} \sin \frac{\psi}{2}, & -\cos \frac{\theta}{2} \sin \frac{\phi}{2} \sin \frac{\psi}{2} + \sin \frac{\theta}{2} \cos \frac{\phi}{2} \cos \frac{\psi}{2}, \\ \cos \frac{\theta}{2} \cos \frac{\phi}{2} \sin \frac{\psi}{2} - \sin \frac{\theta}{2} \sin \frac{\phi}{2} \cos \frac{\psi}{2}, & \cos \frac{\theta}{2} \cos \frac{\phi}{2} \cos \frac{\psi}{2} + \sin \frac{\theta}{2} \sin \frac{\phi}{2} \sin \frac{\psi}{2} \end{pmatrix} \quad (109)$$

となる。(108) 式と (109) 式を比較すると, $q_1 = -q'_2$, $q_2 = -q'_3$, $q_3 = q'_1$, $q_0 = q'_0$ であることが分かる.

この計算を踏まえて先程の話に戻すと, `tf::createQuaternionMsgFromRollPitchYaw()` にオイラー角 $\psi, -\phi, -\theta$ (順番に注意) を渡すことで, (108) 式のクォータニオン (q'_1, q'_2, q'_3, q'_0) が得られる. このクォータニオンから新たなクォータニオン $(-q'_2, -q'_3, q'_1, q'_0)$ を作成すると, これは (109) 式に等しいから, (1) Z 軸周りに ψ , (2) X 軸周りに ϕ , (3) Y 軸周りに θ の順で回転させる. 従って, 当初求めていた回転が得られる. `publishResult()` の `geoQuat` は, (108) 式のクォータニオン (q'_1, q'_2, q'_3, q'_0) に対応し, 順序や符号を入れ替えて作成した (109) 式のクォータニオン $(-q'_2, -q'_3, q'_1, q'_0) = (q_1, q_2, q_3, q_0)$ を `orientation` メンバに格納している. 同様のオイラー角とクォータニオンの処理は, `LaserMapping` の `laserOdometryHandler()` や `publishResult()`, `TransformMaintenance` の `laserOdometryHandler()` や `odomAftMappedHandler()` にも出現する. `position` メンバには `transformSum()` のメンバをそのまま代入する (`LOAM` の座標軸における X, Y, Z 座標).

上記のようにして, 時刻 t_{k+2} における LiDAR の姿勢 `transformSum()` から, `nav_msgs::Odometry` 型のメッセージ `_laserOdometryMsg` を作成し, `/laser_odom_to_init` トピックを通じて Publish する. また `transformSum()` を基に, フレーム間の座標関係を `/tf` トピックに Broadcast している (`RViz` での可視化などに用いる). `tf::StampedTransform` 型の変数 `_laserOdometryTrans` は, 親フレームの ID, 子フレームの ID, フレーム間の姿勢 (並進と回転), タイムスタンプなどの情報を保持しており, ある時点において, 親フレームからみた子フレームの姿勢を表現する. `transformSum()` は `camera_init` フレーム (グローバル座標系) からみた, `laser_odom` フレーム (LiDAR 座標系) の姿勢を表すので, 親フレームの ID を `camera_init`, 子フレームの ID を `laser_odom` に設定し, 回転と並進成分を `setRotation()` および `setOrigin()` によって格納すればよい. この際, オイラー角とクォータニオンの順序に配慮する.

`publishResult()` では最後に, それほど顕著ではないコーナー, それほど顕著ではない平面として分類された点群 `lastCornerCloud()`, `lastSurfaceCloud()` を, `/laser_cloud_corner_last` トピックと, `/laser_cloud_surf_last` トピックから Publish する. また全ての点を含んだ点群 `laserCloud()` を `-velodyne_cloud_3` トピックから Publish する. 先述の通り, これらの点群は, いずれも $k+1$ 番目に取得されたもの (スキャン開始時刻は t_{k+1}) であり, 点の座標はスキャン終了時刻 t_{k+2} (次のスイープの開始時刻, あるいは次の点群のスキャン開始時刻) に揃えられている. 点の座標を t_{k+2} に揃えるのは, `BasicLaserOdometry` の `transformToEnd()` である.

7 BasicLaserMapping クラス

7.1 BasicLaserMapping::transformAssociateMap()

このメソッドは `BasicLaserMapping` の `process()` から最初に呼び出されるので, その状況を前提として説明する. `_transformSum`, `_transformBefMapped`, `_transformAftMapped` が表す姿勢は, T_{k+1}^S, T_k^S, T_k^W で表される. 即ち `_transformBefMapped` と `_transformSum(T_{k+1}^S)` は, オドメトリによって得られた, 時刻 t_{k+1}, t_{k+2} における LiDAR の姿勢である (`_transformBefMapped` は以前の `_transformSum`). `_transformAftMapped` は, マッピングによって得られた, 時刻 t_{k+1} における LiDAR の姿勢である. これら 3 つは全て,

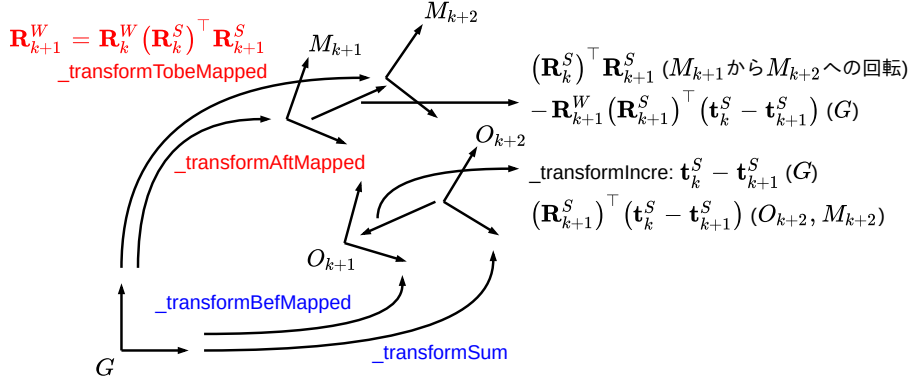
グローバル座標系で表される。図 11 は、transformAssociateMap() で考慮される座標系と、その間の変換をまとめたものである。時刻 t_{k+1}, t_{k+2} における、オドメトリ後の LiDAR 座標系 (/laser_odom フレーム) を O_{k+1}, O_{k+2} , マッピング後の LiDAR 座標系 (/aft_mapped フレーム) を M_{k+1}, M_{k+2} とおく。オドメトリでは、隣接する時刻にスキャンされた点群間、即ち現在の点群と 1 つ前の点群間のレジストレーションによって姿勢を更新する。一方マッピングでは、以前取得された複数の点群から作成した地図に、現在の点群をレジストレーションすることで姿勢を更新するから、オドメトリよりも誤差の累積が少ない。LOAM の実行を開始してしばらくは、オドメトリ後の LiDAR の姿勢 (座標系 O) と、マッピング後の LiDAR の姿勢 (座標系 M) は互いに近い位置にあるが、オドメトリ側に誤差が急速に累積するに従って、両者の姿勢の差は徐々に拡大していく。先ほどの T_k^S, T_{k+1}^S, T_k^W は、座標系 G (グローバル座標系, camera_init フレーム) からみた、座標系 $O_{k+1}, O_{k+2}, M_{k+1}$ の相対的な姿勢である。

T_k^S, T_{k+1}^S, T_k^W の回転成分が、オイラー角で $(\theta_x^k, \theta_y^k, \theta_z^k), (\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}), (\theta_x^{k,W}, \theta_y^{k,W}, \theta_z^{k,W})$ で表されるとする。LOAM では Z, X, Y 軸の順に回転させるから、これらのオイラー角に対応する回転行列は $R_k^S = R_y(\theta_y^k)R_x(\theta_x^k)R_z(\theta_z^k)$, $R_{k+1}^S = R_y(\theta_y^{k+1})R_x(\theta_x^{k+1})R_z(\theta_z^{k+1})$, $R_k^W = R_y(\theta_y^{k,W})R_x(\theta_x^{k,W})R_z(\theta_z^{k,W})$ と書ける。このとき、これら 3 つを合成した回転行列 $R_{k+1}^{W,S} = R_k^W (R_k^S)^\top R_{k+1}^S$ に対応するオイラー角 $\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}$ を求める。この回転は、座標系 G からみた M_{k+2} の相対的な回転、言い換えると、マッピングによって得られた、時刻 t_{k+2} における LiDAR の姿勢 (回転成分) を表す。 $(R_k^S)^\top R_{k+1}^S$ は、座標系 G から O_{k+2}, O_{k+1} から G への回転を合成したものだから、座標系 O_{k+1} から O_{k+2} への回転を表す (O_{k+2} 座標系で表される点の座標を、 O_{k+2} から G , そして G から O_{k+1} へと移すので、座標系を O_{k+1} から O_{k+2} へと回転させる)。この O_{k+1} から O_{k+2} への回転は、オドメトリによって得られた、時刻 t_{k+1} から t_{k+2} にかけての LiDAR の回転である。

マッピングによって得られた、最初の時刻 t_1 から時刻 t_{k+1} までの LiDAR の回転 (座標系 G から M_{k+1} への回転) R_k^W に、オドメトリによって得られた、時刻 t_{k+1} から次の時刻 t_{k+2} までの回転 (座標系 O_{k+1} から O_{k+2} への回転) $(R_k^S)^\top R_{k+1}^S$ を貼り合わせることによって、図 11 の上部のように、最初の時刻 t_1 から時刻 t_{k+2} までの LiDAR の回転 (座標系 G から M_{k+2} への回転) が暫定的に得られる。回転 $(R_k^S)^\top R_{k+1}^S$ はオドメトリ後の座標系 O_{k+1} から O_{k+2} までの回転であるが、これをマッピング後の座標系 M_{k+1} から M_{k+2} として捉えることで、マッピングとオドメトリ双方の結果を結合している。新たに得た回転 $R_{k+1}^{W,S} = R_k^W (R_k^S)^\top R_{k+1}^S$ はマッピング時の初期値となって、地図と点群とのレジストレーションによって改良され、最終的には座標系 G からみた、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 (M_{k+2}) の回転 (姿勢 T_{k+1}^W の回転成分) を表すこととなる。この回転は transformToBeMapped に格納される。

$R_{k+1}^{W,S}$ の回転成分 $\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}$ は次のように求められる。基本的には、BasicLaserOdometry の pluginIMURotation() と同様の流れとなる。行列 R_k^S, R_{k+1}^S, R_k^W の各要素を次のように書き下す。

$$\begin{aligned}
 (R_k^S)^\top &= \begin{bmatrix} \cos \theta_y^k \cos \theta_z^k + \sin \theta_y^k \sin \theta_x^k \sin \theta_z^k & & & \\ -\cos \theta_y^k \sin \theta_z^k + \sin \theta_y^k \sin \theta_x^k \cos \theta_z^k & & & \\ \sin \theta_y^k \cos \theta_x^k & & & \\ & \cos \theta_x^k \sin \theta_z^k & -\sin \theta_y^k \cos \theta_z^k + \cos \theta_y^k \sin \theta_x^k \sin \theta_z^k & \\ & \cos \theta_x^k \cos \theta_z^k & \sin \theta_y^k \sin \theta_z^k + \cos \theta_y^k \sin \theta_x^k \cos \theta_z^k & \\ & -\sin \theta_x^k & \cos \theta_y^k \cos \theta_x^k & \end{bmatrix} \\
 R_{k+1}^S &= \begin{bmatrix} \cos \theta_y^{k+1} \cos \theta_z^{k+1} + \sin \theta_y^{k+1} \sin \theta_x^{k+1} \sin \theta_z^{k+1} & & & \\ & \cos \theta_x^{k+1} \sin \theta_z^{k+1} & & \\ -\sin \theta_y^{k+1} \cos \theta_z^{k+1} + \cos \theta_y^{k+1} \sin \theta_x^{k+1} \sin \theta_z^{k+1} & & & \\ & \cos \theta_x^{k+1} \cos \theta_z^{k+1} & \sin \theta_y^{k+1} \sin \theta_z^{k+1} + \cos \theta_y^{k+1} \sin \theta_x^{k+1} \cos \theta_z^{k+1} & \\ & -\sin \theta_x^{k+1} & \cos \theta_y^{k+1} \cos \theta_x^{k+1} & \end{bmatrix}
 \end{aligned} \tag{110}$$



G : グローバル座標系

O_{k+1} : オドメトリによって得た, 時刻 t_{k+1} におけるLiDAR座標系

O_{k+2} : オドメトリによって得た, 時刻 t_{k+2} におけるLiDAR座標系

M_{k+1} : マッピングによって得た, 時刻 t_{k+1} におけるLiDAR座標系

M_{k+2} : マッピングによって得た, 時刻 t_{k+2} におけるLiDAR座標系

$$\mathbf{T}_k^S = [t_x^k, t_y^k, t_z^k, \theta_x^k, \theta_y^k, \theta_z^k]^\top$$

$$\mathbf{R}_k^S = \mathbf{R}_y(\theta_y^k) \mathbf{R}_x(\theta_x^k) \mathbf{R}_z(\theta_z^k)$$

$$\mathbf{t}_k^S = [t_x^k, t_y^k, t_z^k]^\top$$

$$_transformSum: \mathbf{T}_{k+1}^S (\mathbf{R}_{k+1}^S, \mathbf{t}_{k+1}^S)$$

$$_transformBefMapped: \mathbf{T}_k^S (\mathbf{R}_k^S, \mathbf{t}_k^S)$$

$$_transformAftMapped: \mathbf{T}_k^W (\mathbf{R}_k^W, \mathbf{t}_k^W)$$

$$_transformTobeMapped: \mathbf{T}_{k+1}^W (\mathbf{R}_{k+1}^W, \mathbf{t}_{k+1}^W)$$

$$\mathbf{T}_k^W = [t_x^{k,W}, t_y^{k,W}, t_z^{k,W}, \theta_x^{k,W}, \theta_y^{k,W}, \theta_z^{k,W}]^\top$$

$$\mathbf{R}_k^W = \mathbf{R}_y(\theta_y^{k,W}) \mathbf{R}_x(\theta_x^{k,W}) \mathbf{R}_z(\theta_z^{k,W})$$

$$\mathbf{t}_k^W = [t_x^{k,W}, t_y^{k,W}, t_z^{k,W}]^\top$$

図 11 transformAssociateMap() における座標系

$$\mathbf{R}_k^W = \begin{bmatrix} -\cos \theta_y^{k+1} \sin \theta_z^{k+1} + \sin \theta_y^{k+1} \sin \theta_x^{k+1} \cos \theta_z^{k+1} & \sin \theta_y^{k+1} \cos \theta_x^{k+1} & \cos \theta_x^{k+1} \cos \theta_z^{k+1} \\ \sin \theta_y^{k+1} \sin \theta_z^{k+1} + \cos \theta_y^{k+1} \sin \theta_x^{k+1} \cos \theta_z^{k+1} & -\sin \theta_x^{k+1} & \cos \theta_y^{k+1} \cos \theta_z^{k+1} \\ \cos \theta_y^{k,W} \cos \theta_z^{k,W} + \sin \theta_y^{k,W} \sin \theta_x^{k,W} \sin \theta_z^{k,W} & \cos \theta_x^{k,W} \sin \theta_z^{k,W} & \sin \theta_x^{k,W} \sin \theta_z^{k,W} \\ -\sin \theta_y^{k,W} \cos \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^{k,W} \sin \theta_z^{k,W} & -\cos \theta_y^{k,W} \sin \theta_z^{k,W} + \sin \theta_y^{k,W} \sin \theta_x^{k,W} \cos \theta_z^{k,W} & \sin \theta_y^{k,W} \cos \theta_x^{k,W} \\ \cos \theta_x^{k,W} \cos \theta_z^{k,W} & -\sin \theta_x^{k,W} & \cos \theta_y^{k,W} \cos \theta_x^{k,W} \\ \sin \theta_y^{k,W} \sin \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^{k,W} \cos \theta_z^{k,W} & \cos \theta_y^{k,W} \cos \theta_x^{k,W} & \cos \theta_x^{k,W} \cos \theta_z^{k,W} \end{bmatrix} \quad (111)$$

$$\mathbf{R}_k^W = \begin{bmatrix} \cos \theta_y^{k,W} \cos \theta_z^{k,W} + \sin \theta_y^{k,W} \sin \theta_x^{k,W} \sin \theta_z^{k,W} & \cos \theta_x^{k,W} \sin \theta_z^{k,W} & \sin \theta_x^{k,W} \sin \theta_z^{k,W} \\ -\sin \theta_y^{k,W} \cos \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^{k,W} \sin \theta_z^{k,W} & -\cos \theta_y^{k,W} \sin \theta_z^{k,W} + \sin \theta_y^{k,W} \sin \theta_x^{k,W} \cos \theta_z^{k,W} & \sin \theta_y^{k,W} \cos \theta_x^{k,W} \\ \cos \theta_x^{k,W} \cos \theta_z^{k,W} & -\sin \theta_x^{k,W} & \cos \theta_y^{k,W} \cos \theta_x^{k,W} \\ \sin \theta_y^{k,W} \sin \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^{k,W} \cos \theta_z^{k,W} & \cos \theta_y^{k,W} \cos \theta_x^{k,W} & \cos \theta_x^{k,W} \cos \theta_z^{k,W} \end{bmatrix} \quad (112)$$

行列 \mathbf{R}_{k+1}^W からオイラー角を取り出すには, 行列 \mathbf{R}_{k+1}^W の 5 つの要素 $\mathbf{R}_{k+1,1,12}^W, \mathbf{R}_{k+1,0,2}^W, \mathbf{R}_{k+1,2,2}^W, \mathbf{R}_{k+1,1,10}^W, \mathbf{R}_{k+1,1,11}^W$ を求める必要がある. オイラー角 $\theta_x^{k+1,W}$ を求めるために, 最初に (1, 2) 要素 $\mathbf{R}_{k+1,1,12}^W$ を計算する.

$$\begin{aligned} \mathbf{R}_{k+1,1,12}^W &= (\mathbf{R}_{k,10}^W \mathbf{R}_{k,00}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,01}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,02}^S) \mathbf{R}_{k+1,0,2}^S \\ &\quad (\mathbf{R}_{k,10}^W \mathbf{R}_{k,10}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,11}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,12}^S) \mathbf{R}_{k+1,1,12}^S \\ &\quad (\mathbf{R}_{k,10}^W \mathbf{R}_{k,20}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,21}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,22}^S) \mathbf{R}_{k+1,2,2}^S \\ &= ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (\cos \theta_y^k \cos \theta_z^k + \sin \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\ &\quad + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (-\cos \theta_y^k \sin \theta_z^k + \sin \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\ &\quad + (-\sin \theta_x^{k,W}) (\sin \theta_y^k \cos \theta_x^k)) (\sin \theta_y^{k+1} \cos \theta_x^{k+1}) \\ &\quad + ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (\cos \theta_x^k \sin \theta_z^k) \end{aligned}$$

$$\begin{aligned}
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (\cos \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (-\sin \theta_x^k) (-\sin \theta_x^{k+1}) \\
& + ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (-\sin \theta_y^k \cos \theta_z^k + \cos \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (\sin \theta_y^k \sin \theta_z^k + \cos \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (\cos \theta_y^k \cos \theta_x^k) (\cos \theta_y^{k+1} \cos \theta_x^{k+1})
\end{aligned}$$

続いて、ピッチ角 $\theta_y^{k+1,W}$ を求めるために、 $(0, 2), (2, 2)$ 要素 $\mathbf{R}_{k+1,02}^W, \mathbf{R}_{k+1,22}^W$ を計算する。

$$\begin{aligned}
\mathbf{R}_{k+1,02}^W &= (\mathbf{R}_{k,00}^W \mathbf{R}_{k,00}^S + \mathbf{R}_{k,01}^W \mathbf{R}_{k,01}^S + \mathbf{R}_{k,02}^W \mathbf{R}_{k,02}^S) \mathbf{R}_{k+1,02}^S \\
& \quad (\mathbf{R}_{k,00}^W \mathbf{R}_{k,10}^S + \mathbf{R}_{k,01}^W \mathbf{R}_{k,11}^S + \mathbf{R}_{k,02}^W \mathbf{R}_{k,12}^S) \mathbf{R}_{k+1,12}^S \\
& \quad (\mathbf{R}_{k,00}^W \mathbf{R}_{k,20}^S + \mathbf{R}_{k,01}^W \mathbf{R}_{k,21}^S + \mathbf{R}_{k,02}^W \mathbf{R}_{k,22}^S) \mathbf{R}_{k+1,22}^S \\
&= ((\cos \theta_y^{k,W} \cos \theta_z^{k,W}) (\cos \theta_y^k \cos \theta_z^k + \sin \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& \quad + (-\cos \theta_y^{k,W} \sin \theta_z^{k,W} + \sin \theta_y^{k,W} \sin \theta_x^k \cos \theta_z^{k,W}) (-\cos \theta_y^k \sin \theta_z^k + \sin \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& \quad + (\sin \theta_y^{k,W} \cos \theta_x^{k,W}) (\sin \theta_y^k \cos \theta_x^k) (\sin \theta_y^{k+1} \cos \theta_x^{k+1}) \\
& + ((\cos \theta_y^{k,W} \cos \theta_z^{k,W}) (\cos \theta_x^k \sin \theta_z^k) \\
& \quad + (-\cos \theta_y^{k,W} \sin \theta_z^{k,W} + \sin \theta_y^{k,W} \sin \theta_x^k \cos \theta_z^{k,W}) (\cos \theta_x^k \cos \theta_z^k) \\
& \quad + (\sin \theta_y^{k,W} \cos \theta_x^{k,W}) (-\sin \theta_x^k) (-\sin \theta_x^{k+1}) \\
& + ((\cos \theta_y^{k,W} \cos \theta_z^{k,W}) (-\sin \theta_y^k \cos \theta_z^k + \cos \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& \quad + (-\cos \theta_y^{k,W} \sin \theta_z^{k,W} + \sin \theta_y^{k,W} \sin \theta_x^k \cos \theta_z^{k,W}) (\sin \theta_y^k \sin \theta_z^k + \cos \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& \quad + (\sin \theta_y^{k,W} \cos \theta_x^{k,W}) (\cos \theta_y^k \cos \theta_x^k) (\cos \theta_y^{k+1} \cos \theta_x^{k+1})
\end{aligned}$$

$$\begin{aligned}
\mathbf{R}_{k+1,22}^W &= (\mathbf{R}_{k,20}^W \mathbf{R}_{k,00}^S + \mathbf{R}_{k,21}^W \mathbf{R}_{k,01}^S + \mathbf{R}_{k,22}^W \mathbf{R}_{k,02}^S) \mathbf{R}_{k+1,02}^S \\
& \quad (\mathbf{R}_{k,20}^W \mathbf{R}_{k,10}^S + \mathbf{R}_{k,21}^W \mathbf{R}_{k,11}^S + \mathbf{R}_{k,22}^W \mathbf{R}_{k,12}^S) \mathbf{R}_{k+1,12}^S \\
& \quad (\mathbf{R}_{k,20}^W \mathbf{R}_{k,20}^S + \mathbf{R}_{k,21}^W \mathbf{R}_{k,21}^S + \mathbf{R}_{k,22}^W \mathbf{R}_{k,22}^S) \mathbf{R}_{k+1,22}^S \\
&= ((-\sin \theta_y^{k,W} \cos \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^k \sin \theta_z^{k,W}) (\cos \theta_y^k \cos \theta_z^k + \sin \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& \quad + (\sin \theta_y^{k,W} \sin \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^k \cos \theta_z^{k,W}) (-\cos \theta_y^k \sin \theta_z^k + \sin \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& \quad + (\cos \theta_y^{k,W} \cos \theta_x^{k,W}) (\sin \theta_y^k \cos \theta_x^k) (\sin \theta_y^{k+1} \cos \theta_x^{k+1}) \\
& + ((-\sin \theta_y^{k,W} \cos \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^k \sin \theta_z^{k,W}) (\cos \theta_x^k \sin \theta_z^k) \\
& \quad + (\sin \theta_y^{k,W} \sin \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^k \cos \theta_z^{k,W}) (\cos \theta_x^k \cos \theta_z^k) \\
& \quad + (\cos \theta_y^{k,W} \cos \theta_x^{k,W}) (-\sin \theta_x^k) (-\sin \theta_x^{k+1}) \\
& + ((-\sin \theta_y^{k,W} \cos \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^k \sin \theta_z^{k,W}) (-\sin \theta_y^k \cos \theta_z^k + \cos \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& \quad + (\sin \theta_y^{k,W} \sin \theta_z^{k,W} + \cos \theta_y^{k,W} \sin \theta_x^k \cos \theta_z^{k,W}) (\sin \theta_y^k \sin \theta_z^k + \cos \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& \quad + (\cos \theta_y^{k,W} \cos \theta_x^{k,W}) (\cos \theta_y^k \cos \theta_x^k) (\cos \theta_y^{k+1} \cos \theta_x^{k+1})
\end{aligned}$$

最後に、ヨー角 $\theta_z^{k+1,W}$ を求めるために、 $(1, 0), (1, 1)$ 要素 $\mathbf{R}_{k+1,10}^W, \mathbf{R}_{k+1,11}^W$ を計算する。

$$\begin{aligned}
\mathbf{R}_{k+1,10}^W &= (\mathbf{R}_{k,10}^W \mathbf{R}_{k,00}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,01}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,02}^S) \mathbf{R}_{k+1,00}^S \\
& \quad (\mathbf{R}_{k,10}^W \mathbf{R}_{k,10}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,11}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,12}^S) \mathbf{R}_{k+1,10}^S \\
& \quad (\mathbf{R}_{k,10}^W \mathbf{R}_{k,20}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,21}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,22}^S) \mathbf{R}_{k+1,20}^S \\
&= ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (\cos \theta_y^k \cos \theta_z^k + \sin \theta_y^k \sin \theta_x^k \sin \theta_z^k)
\end{aligned}$$

$$\begin{aligned}
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (-\cos \theta_y^k \sin \theta_z^k + \sin \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (\sin \theta_y^k \cos \theta_x^k) (\cos \theta_y^{k+1} \cos \theta_z^{k+1} + \sin \theta_y^{k+1} \sin \theta_x^{k+1} \sin \theta_z^{k+1}) \\
& + ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (\cos \theta_x^k \sin \theta_z^k) \\
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (\cos \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (-\sin \theta_x^k)) (\cos \theta_x^{k+1} \sin \theta_z^{k+1}) \\
& + ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (-\sin \theta_y^k \cos \theta_z^k + \cos \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (\sin \theta_y^k \sin \theta_z^k + \cos \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (\cos \theta_y^k \cos \theta_x^k)) (-\sin \theta_y^{k+1} \cos \theta_z^{k+1} + \cos \theta_y^{k+1} \sin \theta_x^{k+1} \sin \theta_z^{k+1})
\end{aligned}$$

$$\begin{aligned}
\mathbf{R}_{k+1,11}^W &= (\mathbf{R}_{k,10}^W \mathbf{R}_{k,00}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,01}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,02}^S) \mathbf{R}_{k+1,01}^S \\
& (\mathbf{R}_{k,10}^W \mathbf{R}_{k,10}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,11}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,12}^S) \mathbf{R}_{k+1,11}^S \\
& (\mathbf{R}_{k,10}^W \mathbf{R}_{k,20}^S + \mathbf{R}_{k,11}^W \mathbf{R}_{k,21}^S + \mathbf{R}_{k,12}^W \mathbf{R}_{k,22}^S) \mathbf{R}_{k+1,21}^S \\
&= ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (\cos \theta_y^k \cos \theta_z^k + \sin \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (-\cos \theta_y^k \sin \theta_z^k + \sin \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (\sin \theta_y^k \cos \theta_x^k)) (-\cos \theta_y^{k+1} \sin \theta_z^{k+1} + \sin \theta_y^{k+1} \sin \theta_x^{k+1} \cos \theta_z^{k+1}) \\
& + ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (\cos \theta_x^k \sin \theta_z^k) \\
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (\cos \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (-\sin \theta_x^k)) (\cos \theta_x^{k+1} \cos \theta_z^{k+1}) \\
& + ((\cos \theta_x^{k,W} \sin \theta_z^{k,W}) (-\sin \theta_y^k \cos \theta_z^k + \cos \theta_y^k \sin \theta_x^k \sin \theta_z^k) \\
& + (\cos \theta_x^{k,W} \cos \theta_z^{k,W}) (\sin \theta_y^k \sin \theta_z^k + \cos \theta_y^k \sin \theta_x^k \cos \theta_z^k) \\
& + (-\sin \theta_x^{k,W}) (\cos \theta_y^k \cos \theta_x^k)) (\sin \theta_y^{k+1} \sin \theta_z^{k+1} + \cos \theta_y^{k+1} \sin \theta_x^{k+1} \cos \theta_z^{k+1})
\end{aligned}$$

これらを基に、オイラー角 $\theta_x^{k+1,W}$, $\theta_y^{k+1,W}$, $\theta_z^{k+1,W}$ を次のように復元できる。オイラー角 (\mathbf{T}_{k+1}^W の回転成分) が `_transformTobeMapped` に格納される。

$$\theta_x^{k+1,W} = -\arcsin \mathbf{R}_{k+1,12}^W, \quad \theta_y^{k+1,W} = \arctan \left(\frac{\mathbf{R}_{k+1,02}^W}{\mathbf{R}_{k+1,22}^W} \right), \quad \theta_z^{k+1,W} = \arctan \left(\frac{\mathbf{R}_{k+1,10}^W}{\mathbf{R}_{k+1,11}^W} \right) \quad (113)$$

ソースコード内の変数とは次のように対応している。

$$\begin{aligned}
\text{sbcx} &= \sin \theta_x^{k+1}, & \text{cbcx} &= \cos \theta_x^{k+1}, & \text{sbcy} &= \sin \theta_y^{k+1}, & \text{cbcy} &= \cos \theta_y^{k+1}, & \text{sbcz} &= \sin \theta_z^{k+1}, & \text{cbcz} &= \cos \theta_z^{k+1} \\
\text{sblx} &= \sin \theta_x^k, & \text{cblx} &= \cos \theta_x^k, & \text{sby} &= \sin \theta_y^k, & \text{cby} &= \cos \theta_y^k, & \text{sblz} &= \sin \theta_z^k, & \text{cblz} &= \cos \theta_z^k \\
\text{salx} &= \sin \theta_x^{k,W}, & \text{calx} &= \cos \theta_x^{k,W}, & \text{saly} &= \sin \theta_y^{k,W}, & \text{caly} &= \cos \theta_y^{k,W}, & \text{salz} &= \sin \theta_z^{k,W}, & \text{calz} &= \cos \theta_z^{k,W}
\end{aligned}$$

$$\begin{aligned}
\text{srx} &= \mathbf{R}_{k+1,12}^W, & \text{acx} &= \theta_x^{k+1,W} \\
\text{srycrx} &= \mathbf{R}_{k+1,02}^W, & \text{crycrx} &= \mathbf{R}_{k+1,22}^W, & \text{acy} &= \theta_y^{k+1,W} \\
\text{srzcrx} &= \mathbf{R}_{k+1,10}^W, & \text{crzcrx} &= \mathbf{R}_{k+1,11}^W, & \text{acz} &= \theta_z^{k+1,W}
\end{aligned}$$

姿勢 \mathbf{T}_{k+1}^W の回転成分 $\theta_x^{k+1,W}$, $\theta_y^{k+1,W}$, $\theta_z^{k+1,W}$ は得たので、続いて並進成分 $\mathbf{t}_{k+1}^W = [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^\top$ を求めよう。この t_{k+1}^W は、図 11 に示すグローバル座標系 G からみた、時刻 t_{k+1} におけるマッピング後の座標系 M_{k+2} の位置である。回転行列の場合と同様に、最初の時刻 t_1 から時刻 t_{k+1} までのマッピングの結果と、

時刻 t_{k+1} から t_{k+2} までのオドメトリの結果を繋ぎ合わせることで求める。座標系 O_{k+2} から O_{k+1} への位置の変化 $t_k^S - t_{k+1}^S$ を `_transformIncre` に格納する ($t_k^S = [t_x^k, t_y^k, t_z^k]^\top$, $t_{k+1}^S = [t_x^{k+1}, t_y^{k+1}, t_z^{k+1}]^\top$)。 t_k^S (`_transformBefMapped`), t_{k+1}^S (`_transformSum`) はグローバル座標系で表されるから、その差 $t_k^S - t_{k+1}^S$ もグローバル座標系である。これに $(R_{k+1}^S)^\top$ を適用することで、グローバル座標系 G から O_{k+2} 座標系へと移す。 $(R_{k+1}^S)^\top$ は、 O_{k+2} 座標系からみたグローバル座標系 G の回転であるから、点の座標をグローバル座標系 G から O_{k+2} 座標系へと変換する。最初の `rotateXYZ()` の呼び出し時点では、`_transformIncre` には $(R_{k+1}^S)^\top (t_k^S - t_{k+1}^S)$ が格納されている。

$$\begin{aligned} (R_{k+1}^S)^\top (t_k^S - t_{k+1}^S) &= (R_y(\theta_y^{k+1})R_x(\theta_x^{k+1})R_z(\theta_z^{k+1}))^\top (t_k^S - t_{k+1}^S) \\ &= (R_z(\theta_z^{k+1})^\top R_x(\theta_x^{k+1})^\top R_y(\theta_y^{k+1})^\top) (t_k^S - t_{k+1}^S) \end{aligned} \quad (114)$$

`rotateXYZ()` は Y, X, Z 軸の順に回転させる関数であり、上式のようにオイラー角の符号を反転させてから与える。

この時点での `_transformIncre` は O_{k+2} 座標系であるが、これを M_{k+2} 座標系とみなして、時刻 t_1 から t_{k+1} までのマッピングの結果 $t_k^W = [t_x^{k,W}, t_y^{k,W}, t_z^{k,W}]^\top$ (`_transformAftMapped` の並進成分) と連結する。 `_transformIncre` は O_{k+2} 座標系からみた O_{k+1} 座標系の位置であるが、 M_{k+2} 座標系からみた M_{k+1} 座標系の位置として捉えるわけである。 t_k^W は、グローバル座標系 G からみた、時刻 t_{k+1} におけるマッピング後の座標系 M_{k+1} の位置である。 t_k^W はグローバル座標系で表されているから、`_transformIncre` を足し合わせる前に、`_transformIncre` の座標を M_{k+2} 座標系からグローバル座標系 G へと再度変換する必要がある。回転 R_{k+1}^W は、グローバル座標系 G からみた M_{k+2} 座標系の回転であり、点の座標を M_{k+2} 座標系からグローバル座標系 G へと移すので、これを適用すればよい。

$$\begin{aligned} R_{k+1}^W (R_{k+1}^S)^\top (t_k^S - t_{k+1}^S) &= (R_k^W (R_k^S)^\top R_{k+1}^S) (R_{k+1}^S)^\top (t_k^S - t_{k+1}^S) \\ &= R_k^W (R_k^S)^\top (t_k^S - t_{k+1}^S) \end{aligned} \quad (115)$$

上で求めたのは、座標系 M_{k+2} から M_{k+1} への位置変化をグローバル座標系で表したものである。 `rotateZXY()` の呼び出しによって、ベクトル v に上記が格納される。その符号を反転させれば、座標系 M_{k+1} から M_{k+2} への位置変化を表す。符号を反転させたベクトル v を、グローバル座標系 G から M_{k+1} への位置変化 t_k^W に足し合わせれば、グローバル座標系 G から M_{k+2} への位置変化 t_{k+1}^W 、即ちグローバル座標系 G からみた、時刻 t_{k+2} におけるマッピング後の座標系 M_{k+2} の位置が得られる。

$$t_{k+1}^W = t_k^W - R_{k+1}^W (R_{k+1}^S)^\top (t_k^S - t_{k+1}^S) \quad (116)$$

上記は、時刻 t_1 から t_{k+1} までのマッピングの結果 t_{k+1}^W に、時刻 t_{k+1} から t_{k+2} までのオドメトリの結果 $-R_{k+1}^W (R_{k+1}^S)^\top$ を足し合わせている。 t_{k+1}^W (T_{k+1}^W の並進成分) が `_transformTobeMapped` に格納される。

以上をまとめると、`transformAssociateMap()` では `_transformSum` ($T_{k+1}^S = [t_{k+1}^S, R_{k+1}^S]$), `_transformBefMapped` ($T_k^S = [t_k^S, R_k^S]$), `_transformAftMapped` ($T_k^W = [t_k^W, R_k^W]$) の3つを使って、`_transformIncre` ($(R_k^S)^\top (t_k^S - t_{k+1}^S)$) と `_transformTobeMapped` ($T_{k+1}^W = [t_{k+1}^W, R_{k+1}^W]$) を計算する。但し $R_{k+1}^W = R_k^W (R_k^S)^\top R_{k+1}^S$, $t_{k+1}^W = t_k^W - R_{k+1}^W (R_{k+1}^S)^\top (t_k^S - t_{k+1}^S)$ である。 `_transformIncre` は、 M_{k+2} 座標系からみた M_{k+1} 座標系の位置、言い換えれば、オドメトリによって得られた、時刻 t_{k+2} から t_{k+1} にかけての LiDAR 座標系の位置変化を表す (符号を反転させれば、時刻 t_{k+1} から t_{k+2} にかけての LiDAR 座標系の位置変化となる)。 `_transformTobeMapped` は、グローバル座標系 G からみた M_{k+2} 座標系の姿勢、即ち、

グローバル座標系 G から、時刻 t_{k+2} におけるマッピング後の座標系 M_{k+2} への姿勢変化を表したものである。但し先述のように、前回のマッピングの結果に、最新のオドメトリの結果を貼り合わせて得られた初期値である。後続のマッピングおよび最適化処理によって、`_transformTobeMapped` は改良される。

7.2 BasicLaserMapping::transformUpdate()

IMU データを使用して、マッピングと最適化処理によって得られた `_transformTobeMapped` を修正する。`_transformTobeMapped` は、グローバル座標系 G からみた、座標系 M_{k+2} の相対的な姿勢 (図 11 参照) であり、 M_{k+2} は時刻 t_{k+2} におけるマッピング後の LiDAR 座標系を指す。

最初に、時刻 t_{k+2} における LiDAR(IMU) の回転、即ちグローバル座標系からみた LiDAR あるいは IMU の回転量 (オイラー角) を IMU データから取得する。リングバッファ `_imuHistory` から、時刻 t_{k+2} の前後で取得された IMU データ (インデックス `imuIdx - 1` と `imuIdx`) を取り出す。この IMU データ `_imuHistory[imuIdx - 1]`、`_imuHistory[imuIdx]` を使って、時刻 t_{k+2} における IMU データ `imuCur` を線形補間によって求める。IMUState2 の `interpolate()` で求めているのは、時刻 t_{k+2} におけるロール角とピッチ角 (ROS の座標軸における \tilde{X} 軸、 \tilde{Y} 軸まわりの回転角) である (ヨー角は利用しない)。時刻 t_{k+2} 以降の IMU データがバッファに存在しない、あるいは IMU データがバッファ内に 1 つしかない場合は線形補間 (内挿) できないので、最新の IMU データ `_imuHistory[imuIdx]` を `imuCur` としてそのまま利用する。時刻 t_{k+2} は、`_laserOdometryTime`(時刻 t_{k+1}) に、点群の取得間隔 `_scanPeriod` を足せば得られる。`imuCur` のピッチ角とロール角を利用して、`_transformTobeMapped` のロール角とヨー角 (LOAM の座標軸における X 軸、 Z 軸まわりの回転角だから、ROS の座標軸における \tilde{Y} 軸、 \tilde{X} 軸まわりの回転角) を修正する。この修正でも線形補間が利用される (`_transformTobeMapped` と `imuCur` のオイラー角を、係数 0.998, 0.002 で重み付けして足し合わせる)。

IMU データのヨー角 (ROS の座標軸における \tilde{Z} 軸まわりの回転) は利用されない。 \tilde{Z} 軸は上方向なので、ヨー角はロボットがどの方向 (東西南北) を向いているかを意味する。3 つのオイラー角のなかでヨー角は最も変化量が大きく (ロボットが部屋を一周する場合は 360° 変化する)、SLAM の性能にも影響を与えるので、IMU による補間は行わないと考えられる (IMU よりもマッピングの結果を信頼する)。

`_transformTobeMapped` を `_transformAftMapped`、`_transformSum` を `_transformBefMapped` に代入することで、`_transformAftMapped` は T_k^W から T_{k+1}^W へ、また `_transformBefMapped` は T_k^S から T_{k+1}^S へとそれぞれ更新される。

7.3 BasicLaserMapping::pointAssociateToMap()

時刻 t_{k+1} の点群に含まれる点の座標 pi が、時刻 t_{k+2} におけるオドメトリ後の座標系 O_{k+2} で表されるとする (図 11 参照)。 M_{k+2} と O_{k+2} はいずれも、同一時刻における LiDAR の座標系であるから、点の座標 pi は、時刻 t_{k+2} におけるマッピング後の座標系 M_{k+2} としても解釈できる。グローバル座標系 G からみた O_{k+2} の姿勢 (オドメトリによって得た LiDAR の姿勢) T_{k+1}^S を使って、座標 pi をグローバル座標に変換するよりも、グローバル座標系 G からみた M_{k+2} の姿勢 (マッピングによって得た LiDAR の姿勢) T_{k+1}^W を使って、座標 pi をグローバル座標に変換した方が正確なはずである (T_{k+1}^S よりも T_{k+1}^W の方が累積誤差が少ない)。

このメソッドでは `_transformTobeMapped` ($T_{k+1}^W = [t_{k+1}^W, R_{k+1}^W]$) を使って、座標 pi を M_{k+2} 座標系からグローバル座標系 G へと変換し、その結果を po に格納する。何度も記述している通り、 T_{k+1}^W はグローバル座

標系 G から M_{k+2} への座標軸の姿勢変化であるから、 M_{k+2} 座標系で表される点の座標をグローバル座標系 G へと変換する。 t_{k+1}^W は並進成分 (位置), R_{k+1}^W は回転行列である。座標 pi を p とすれば、この変換は次のようになる (`rotateZXY()` を使う)。

$$R_{k+1}^W p + t_{k+1}^W = (R_y(\theta_y^{k+1,W}) R_x(\theta_x^{k+1,W}) R_z(\theta_z^{k+1,W})) p + t_{k+1}^W \quad (117)$$

7.4 BasicLaserMapping::pointAssociateToMapped()

`pointAssociateToMap()` とは逆の変換を行う。即ち、グローバル座標系 G で表される点の座標 pi を、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 M_{k+2} に変換する (図 11 参照)。言い換えると、グローバル座標系で pi の位置にある点が、時刻 t_{k+2} の LiDAR からどの位置に見えるかを計算する。 `_transformToMapped($T_{k+1}^W = [t_{k+1}^W, R_{k+1}^W]$)` は、マッピングによって得られた、時刻 t_{k+2} における LiDAR の姿勢、即ちグローバル座標系 G からみた、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 M_{k+2} の相対姿勢を表す。回転行列の転置 (R_{k+1}^W)[⊤] は、 M_{k+2} からグローバル座標系 G への座標軸の姿勢変化であるから、グローバル座標系 G で表される点の座標を M_{k+2} 座標系へと変換する。 t_{k+1}^W は並進成分 (位置), R_{k+1}^W は回転行列である。座標 pi を p とすれば、この変換は次のようになる (`rotateYZX()` を使う)。

$$\begin{aligned} (R_{k+1}^W)^\top (p - t_{k+1}^W) &= (R_y(\theta_y^{k+1,W}) R_x(\theta_x^{k+1,W}) R_z(\theta_z^{k+1,W}))^\top (p - t_{k+1}^W) \\ &= (R_z(\theta_z^{k+1,W})^\top R_x(\theta_x^{k+1,W})^\top R_y(\theta_y^{k+1,W})^\top) (p - t_{k+1}^W) \\ &= (R_z(-\theta_z^{k+1,W}) R_x(-\theta_x^{k+1,W}) R_y(-\theta_y^{k+1,W})) (p - t_{k+1}^W) \end{aligned} \quad (118)$$

7.5 BasicLaserMapping::transformFullResToMap()

現在の ($k+1$ 番目の) 点群 `_laserCloudFullRes` に含まれる各点の座標は、時刻 t_{k+2} におけるオドメトリ後の座標系 O_{k+2} で表される (図 11 参照)。 `pointAssociateToMap()` を使うことで、各点の座標を、時刻 t_{k+2} におけるマッピング後の座標系 M_{k+2} とみなしたうえで、 `_transformToMapped(T_{k+1}^W)` に基づいてグローバル座標系に変換する。

7.6 BasicLaserMapping::createDownsizedMap()

このメソッドは `process()` から 5Hz の頻度で呼び出され、そのうち `_mapFrameCount` 回に 1 回 (デフォルトでは 5 回に 1 回) 実行されるので、結局 1Hz の頻度で実行することになる (デフォルトのパラメータ設定の場合)。LiDAR の周囲 $1050\text{m} \times 550\text{m} \times 1050\text{m}$ の空間は、一辺が 50m のボクセルで仕切って管理され (ボクセルの数は合計 $21 \times 11 \times 21 = 4851$ 個)、各ボクセルには 1 つの小さな点群が割り当てられている。ボクセルの個数は `BasicLaserMapping` の `_laserCloudWidth` (X 軸方向の個数)、 `_laserCloudHeight` (Y 軸方向の個数)、 `_laserCloudDepth` (Z 軸方向の個数) で表され (デフォルトではそれぞれ 21, 11, 21)、ボクセルの総数は `_laserCloudNum` メンバに格納されている (デフォルトでは 4851)。 `_laserCloudCornerArray` と `_laserCloudSurfArray` は `_laserCloudNum` 個の小さな点群 (ボクセル) のリストであり、LiDAR の周囲 $1050\text{m} \times 550\text{m} \times 1050\text{m}$ の空間にある、コーナーあるいは平面として分類された点を管理している。

`_laserCloudSurroundInd` は、LiDAR の周囲 $250\text{m} \times 250\text{m} \times 250\text{m}$ にある、最大 $5^3 = 125$ 個のボクセルのインデックスを格納している (インデックスは 0 から 4850 までの整数)。時刻 t_{k+2} における LiDAR の姿勢は、

グローバル座標系で $T_{k+1}^W = [t_{k+1}^W, R_{t+1}^W]$ として表されるので、 t_{k+1}^W の近くにあるボクセルのインデックスをとればよい。_laserCloudSurroundInd に含まれる各インデックスについて、対応するボクセル (点群) を _laserCloudCornerArray と _laserCloudSurfArray から取り出して、_laserCloudSurround に追加する。これらの点群に含まれる各点の座標は、グローバル座標系で表現される。この作成方法から明らかな通り、_laserCloudSurround は LiDAR の周囲にある点を集めた点群である。点群 _laserCloudSurround にボクセルフィルタ _downSizeFilterCorner を適用して点を間引き、その結果を _laserCloudSurroundDS に格納する (_downSizeFilterMap ではなく _downSizeFilterCorner を適用している)。_laserCloudSurroundDS は 1Hz の頻度で更新され、/laser_cloud_surround トピックから Publish される。

7.7 BasicLaserMapping::process()

LaserMapping ノードの主な処理を行う。現在 ($k+1$ 番目) の点群を、以前の複数の点群から作成された地図にレジストレーションすることで、時刻 t_{k+2} における LiDAR の姿勢を求める。このメソッドも長いので、分割して説明する。

7.7.1 地図と点群の前準備

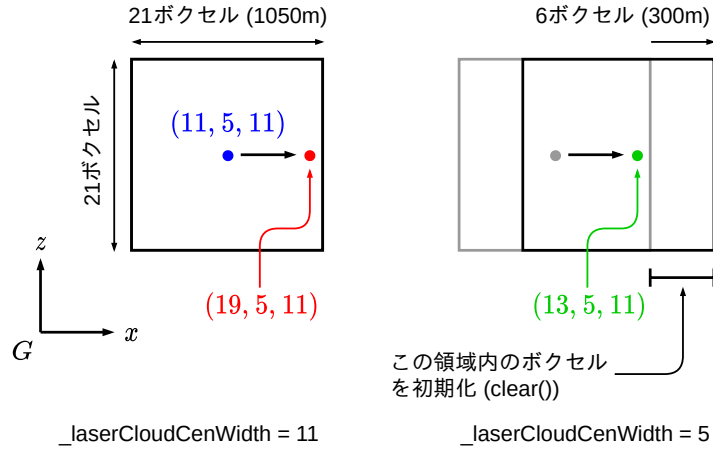
laserOdometryTime には、現在の点群のスキャン開始時刻 laserOdometryTime(t{k+1}) を格納する。_laserOdometryTime は transformUpdate() 内で使用される。

このメソッドが呼び出された時点で、_transformBefMapped と _transformSum には、オドメトリによって得られた時刻 t_{k+1}, t_{k+2} における LiDAR の姿勢 T_k^S, T_{k+1}^S 、_transformAftMapped には、マッピングによって得られた時刻 t_{k+1} における LiDAR の姿勢 T_k^W が格納されている (詳しくは transformAssociateMap() を参照)。transformAssociateMap() を呼び出すことで、オドメトリによって得られた、時刻 t_{k+2} から t_{k+1} にかけての LiDAR の姿勢変化が _transformIncre に、またマッピングによって得られた時刻 t_{k+2} における LiDAR の姿勢 T_{k+1}^W が _transformTobeMapped に格納される。_transformTobeMapped は初期値であり、後続の最適化処理によって更新される。

現在 ($k+1$ 番目) の点群のうち、コーナーあるいは平面として分類された点を集めた点群が _laserCloudCornerLast, _laserCloudSurfLast である。2 つの点群に含まれる各点は、時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 (図 11 の O_{k+2})、あるいは時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 (図 11 の M_{k+2}) で表される。pointAssociateToMap() によって各点の座標 (M_{k+2} 座標系) を、姿勢 T_{k+1}^W (グローバル座標系からみた M_{k+2} 座標系の姿勢) を使ってグローバル座標系へと変換し、_laserCloudCornerStack と _laserCloudSurfStack に格納する。

LaserMapping ノードでは、LiDAR の周囲 $1050\text{m} \times 550\text{m} \times 1050\text{m}$ にある点が、地図として管理される。この領域は一辺 50m のボクセルで区切られ (合計 $21 \times 11 \times 21 = 4851$ 個)、各ボクセルには点群が割り当てられている。LiDAR の移動に伴って、地図として管理すべき $1050\text{m} \times 550\text{m} \times 1050\text{m}$ の領域も徐々に移動する。LiDAR の移動によって、管理対象の領域から外れたボクセル (点群) を削除し、新たに管理対象となった領域に含まれるボクセル (点群) を初期化する必要がある (図 12)。

LiDAR のグローバル座標系での位置 _transformTobeMapped ($t_{k+1}^W = [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^T$) と対応する、ボクセルの X, Y, Z 軸方向のインデックス centerCubeI, centerCubeJ, centerCubeK を最初に計算する。_laserCloudCenWidth, _laserCloudCenHeight, _laserCloudCenDepth は、 X 軸、 Y 軸、 Z 軸方向のインデックスのオフセットで、それぞれ 11, 5, 11 で初期化される。初期状態では、LiDAR の位置はグローバル



- 時刻 t_1 におけるcenterCube{I, J, K}
- $t_1^W = [0, 0, 0]^T$ に対応するインデックス
- 時刻 t_3 におけるcenterCube{I, J, K}
- $t_3^W = [400, 0, 0]^T$ に対応するインデックス
- 時刻 t_3 におけるcenterCube{I, J, K} (スライド後)
- $t_3^W = [400, 0, 0]^T$ に対応するインデックス

図 12 地図として管理される領域

座標系で原点にあるので ($t_1^W = [0, 0, 0]^T$), LiDAR の位置に対応するボクセルのインデックスは (11, 5, 11) となって, $21 \times 11 \times 21$ のボクセルの中心を指している (図 12 の青点).

$t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}$ と centerCubeI, J, K との間には次の関係が成り立っている.

$$\begin{aligned} \text{centerCubeI} &= \text{int} \left((t_x^{k+1,W} + \text{CUBE_HALF}) \div \text{CUBE_SIZE} \right) + _laserCloudCenWidth \\ \text{centerCubeJ} &= \text{int} \left((t_y^{k+1,W} + \text{CUBE_HALF}) \div \text{CUBE_SIZE} \right) + _laserCloudCenHeight \\ \text{centerCubeK} &= \text{int} \left((t_z^{k+1,W} + \text{CUBE_HALF}) \div \text{CUBE_SIZE} \right) + _laserCloudCenDepth \end{aligned}$$

CUBE_SIZE はボクセルの一辺の長さ (50m), CUBE_HALF はその半分 (25m) である. $\text{int}()$ は, 小数点数から小数部分を取り除く関数である (C 言語での float から int のキャスト). 従って, $t_{k+1}^W = [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^T$ が中心に位置するような, ボクセルのインデックスを求めている. 気を付けたいといけないのは, 上記は座標 $t_x^{k+1,W} + \text{CUBE_HALF}, t_y^{k+1,W} + \text{CUBE_HALF}, t_z^{k+1,W} + \text{CUBE_HALF}$ が正のときにのみ成立する. 例えば $_laserCloudCenWidth$ が 0 で $t_x^{k+1,W} = 10\text{m}$ のときは centerCubeI は 0 となって, これは正しい (0 番目のボクセルの左端から $10 + 25 = 35\text{m}$ の位置に LiDAR がある). $t_x^{k+1,W} = -30\text{m}$ のときは, 0 番目のボクセルの左端から $10 - 30 = -20\text{m}$, 言い換えると -1 番目のボクセルの左端から 30m の位置に LiDAR があるから, centerCubeI は -1 とすべきである. しかし上式では $\text{int}((10 - 35) \div 50) = \text{int}(-0.5) = 0$ となるから, centerCubeI は -1 ではなく 0 になってしまう. それでは困るので, 座標 $t_x^{k+1,W} + \text{CUBE_HALF}, t_y^{k+1,W} + \text{CUBE_HALF}, t_z^{k+1,W} + \text{CUBE_HALF}$ が負のときは, 上記で求めたインデックス centerCubeI, J, K から -1 を引いて調節している.

$_laserCloudCenWidth, _laserCloudCenHeight, _laserCloudCenDepth$ のオフセットが必要な理由は次のように説明できる. 初期状態で LiDAR が原点 $t_1^W = [0, 0, 0]^T$ にあり, 次の時刻に $t_2^W = [50, 0, 0]^T$ に移

動したとすると (X 軸正方向にボクセル 1 個分), LiDAR の現在位置 t_2^W に対応するボクセルのインデックス (`centerCubeI`, J , K) は $(12, 5, 11)$ となる ($12 = 50 \div 50 + 11$). LiDAR が更に $t_3^W = [400, 0, 0]^T$ まで移動したとすると (X 軸正方向にボクセル 8 個分), LiDAR の現在位置に対応するインデックス (`centerCubeI`, J , K) は $(19, 5, 11)$ となる ($19 = 400 \div 50 + 11$). このとき, LiDAR の現在位置 t_3^W よりも X 軸負方向にはボクセルが 19 個あるが (インデックス 0 から 18 まで), X 軸正方向にはボクセルが 1 個しかない (インデックス 20 のみ). 従って, 現在の地図 (LiDAR を含む $21 \times 11 \times 21$ 個のボクセル) は, LiDAR の周囲を捉えているとはいえない (特に X 軸方向に偏っている, 図 12 の赤点). LiDAR の現在位置 t_3^W に対応するボクセルが, 再び地図の中心付近に来るように, 地図として管理する領域を, X 軸正方向にスライドさせる必要がある. 例えば, X 軸正方向にボクセル 6 個分 (300m) だけ領域をスライドさせると, `_laserCloudCenWidth` は初期値の 11 から 5 へと更新され, 現在位置 t_3^W に対応するボクセルのインデックス (`centerCubeI`, J , K) は $(13, 5, 11)$ となるから ($13 = 400 \div 50 + 5$), 地図の中心付近に LiDAR が存在するようになる (図 12 の緑点).

このように, 地図として管理する領域が各軸に沿ってスライドするに従って, オフセットは変更される. 領域をスライドさせるのは, LiDAR が地図の中心付近にあって, 地図が LiDAR 周囲の点群を表すようにしたいからである. X 軸正方向にボクセル 1 個分 (50m) だけ領域を移動させたら, X 軸方向のオフセット `_laserCloudCenWidth` から 1 を引く. グローバル座標系で同じ位置に対応するボクセルの, X 方向のインデックスは 1 だけ減る. これは Y 軸, Z 軸方向のオフセット `_laserCloudCenHeight`, `_laserCloudCenDepth` に関しても同様である.

LiDAR の現在位置 `_transformToBeMapped` に対応するボクセルのインデックス `centerCubeI`, J , K を計算したら, そのボクセルが地図のなかで端の方に寄り過ぎていないか調べる. 各インデックスがそれぞれ, 3 から `_laserCloudWidth - 4`, 3 から `_laserCloudHeight - 4`, 3 から `_laserCloudDepth - 4` の範囲に収まっていなければ, LiDAR の現在位置は地図の端の方にある. この場合は, 地図として管理する領域を各軸に沿って適切にスライドさせ, LiDAR の現在位置が地図の中心付近に来るようにする. ソースコード内では似たような for 文が 6 つ連続するが, X, Y, Z 軸に対して負方向, 正方向にスライドさせるためである.

最初の for ループでは, 管理対象の領域を必要な分だけ (`centerCubeI` が 3 以上になるまで), X 軸負方向にスライドさせている. 地図内のボクセル (点群) を X 軸正方向にスライドさせ (`std::swap()` によりインデックス $i - 1$ のボクセルを i へと移す), 新たに管理対象となった領域内のボクセルを初期化している (インデックスが 0 のボクセルに対して `clear()` を呼び出す). また, `centerCubeI` と `_laserCloudCenWidth` をインクリメントする. 次の for ループでは, 領域を必要な分だけ (`centerCubeI` が `_laserCloudWidth - 3` 未満になるまで), X 軸正方向にスライドさせている. 地図内のボクセル (点群) を X 軸負方向にスライドさせ (インデックス $i + 1$ のボクセルを i へと移す), 新たに管理対象となった領域内のボクセルを初期化している (インデックスが `_laserCloudWidth - 1` のボクセルに対して `clear()` を呼び出す). また, `centerCubeI` と `_laserCloudCenWidth` をデクリメントする. 同様の処理を Y, Z 軸に対しても行う.

ここまでで, LiDAR の現在位置 `_transformToBeMapped` に対するボクセルのインデックス `centerCubeI`, J , K と, X, Y, Z 軸方向のオフセット `_laserCloudCenWidth`, `Height`, `Depth` が調節された. 続いて, `centerCubeI`, J , K を中心とする $5 \times 5 \times 5$ のボクセルについて, それが地図として管理される領域の内側かどうか, また各ボクセルが LiDAR の視野に入っているかどうか (`isInLaserFOV`) を調べる. ボクセル i, j, k があるとき, それが地図の内側かどうかは, インデックスが 0 から `_laserCloudWidth - 1`, 0 から `_laserCloudHeight - 1`, 0 から `_laserCloudDepth - 1` の範囲に収まっているかどうかで判断できる. 地図の内側のボクセルについては, それが LiDAR の視野に入っているかどうか (`isInLaserFOV`) を次のように調べる (図 13).

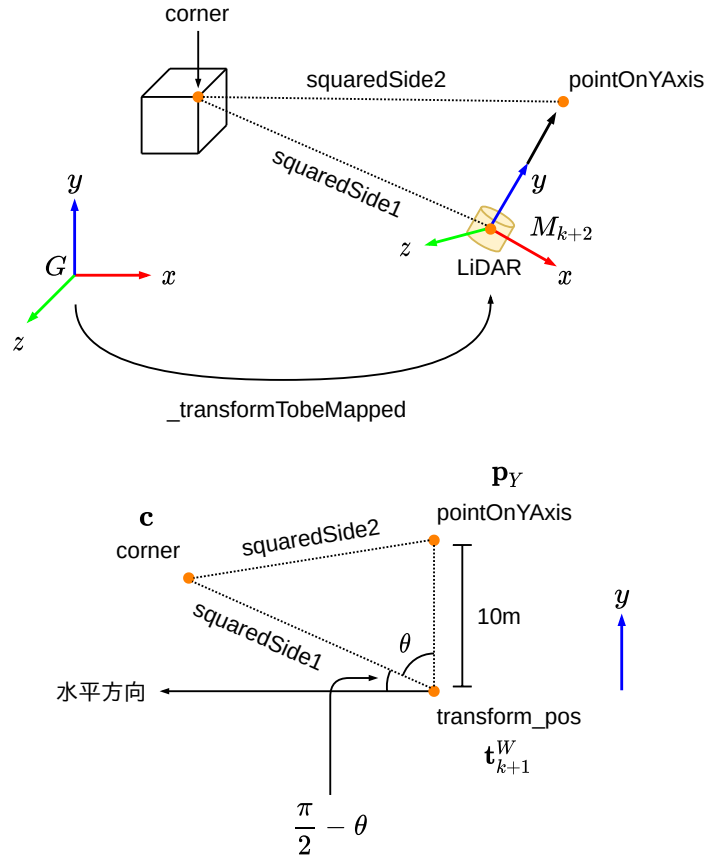


図 13 transformAssociateMap() における FOV(Field Of View) の判定

ボクセルのインデックス i, j, k と、中心位置 $centerX, Y, Z$ (グローバル座標系) との間には次の関係が成り立っている。

$$\begin{aligned}
 i &= \text{centerX} \div \text{CUBE_SIZE} + \text{_laserCloudCenWidth} \\
 j &= \text{centerY} \div \text{CUBE_SIZE} + \text{_laserCloudCenHeight} \\
 k &= \text{centerZ} \div \text{CUBE_SIZE} + \text{_laserCloudCenDepth}
 \end{aligned}$$

中心位置 $centerX, Y, Z$ に CUBE_HALF (25m) を足せば、ボクセル内で最大の座標、 CUBE_HALF を引けばボクセル内で最小の座標となる。上式を使って i, j, k から $centerX, Y, Z$ を求めて、各座標値に $\pm 25\text{m}$ を足すことで、ボクセルの 8 つの頂点の座標 corner を計算する (図 13)。8 つの頂点のどれか 1 つでも LiDAR から見えているなら、ボクセルは見えていると判定する (isInLaserFOV を true にセットする)。LiDAR から 10m だけ真上にある点の座標 $p_Y = [0, 10, 0]^T$ を pointOnYAxis とする (図 13)。この座標は図 11 の M_{k+2} 座標系、即ち時刻 t_{k+2} におけるマッピング後の LiDAR 座標系で表される。LOAM が扱う座標軸では Y 軸が上方向なので、座標の Y 成分を 10 とする。pointAssociateToMap() を使って、 M_{k+2} 座標系の pointOnYAxis をグローバル座標系へと変換しておく。ボクセルの頂点 $c(\text{corner})$ と、LiDAR の位置 t_{k+1}^W ($\text{_transformTobeMapped}$) との距離 $\|c - t_{k+1}^W\|^2$ を squaredSide1 、ボクセルの頂点 $c(\text{corner})$ と、LiDAR の真上にある点 p_Y (pointOnYAxis) との距離 $\|c - p_Y\|^2$ を squaredSide2 に代入する。LiDAR の位置 t_{k+1}^W から、その真上にある点 p_Y までの距離は 10m である。図 13 に示すように、余弦定理から次の関係式が得ら

れる。

$$\|c - t_{k+1}^W\|^2 + 10^2 - 2 \cdot 10 \|c - t_{k+1}^W\| \cos \theta = \|c - p_Y\|^2 \quad (119)$$

θ は、ベクトル $p_Y - t_{k+1}^W$ と $c - t_{k+1}^W$ とのなす角 ($0 \leq \theta \leq \pi$) である。これを变形すると次のようになる。

$$10^2 + \|c - t_{k+1}^W\|^2 - \|c - p_Y\|^2 = 20 \|c - t_{k+1}^W\| \cos \theta \quad (120)$$

ここでは、仰角 ($\pi/2 - \theta$) が -60° から 60° 、言い換えると θ が 30° から 150° のときに、ボクセルの頂点 corner が LiDAR から見えているとする。この条件 $30^\circ < \theta < 150^\circ$ は次のように書き直せる。

$$\begin{aligned} \frac{\pi}{6} < \theta < \frac{5\pi}{6} \\ \Rightarrow -\frac{\sqrt{3}}{2} < \cos \theta < \frac{\sqrt{3}}{2} \\ \Rightarrow -\frac{\sqrt{3}}{2} \cdot 20 \|c - t_{k+1}^W\| < 20 \|c - t_{k+1}^W\| \cos \theta < \frac{\sqrt{3}}{2} \cdot 20 \|c - t_{k+1}^W\| \\ \Rightarrow 20 \|c - t_{k+1}^W\| \cos \theta - 10\sqrt{3} \|c - t_{k+1}^W\| < 0 < 20 \|c - t_{k+1}^W\| \cos \theta + 10\sqrt{3} \|c - t_{k+1}^W\| \\ \Rightarrow 10^2 + \|c - t_{k+1}^W\|^2 - \|c - p_Y\|^2 - 10\sqrt{3} \|c - t_{k+1}^W\| < 0 \quad \text{かつ} \\ 0 < 10^2 + \|c - t_{k+1}^W\|^2 - \|c - p_Y\|^2 + 10\sqrt{3} \|c - t_{k+1}^W\| \end{aligned} \quad (121)$$

ソースコード内では、変数 check1 を使って 1 つ目の条件 (< 0)、変数 check2 を使って 2 つ目の条件 (> 0) をチェックしている。ボクセルの頂点の座標 corner が両方の条件を満たせば、その頂点は LiDAR から -60° 以上 60° 未満の仰角で観測され、ボクセルは LiDAR の視野に入っていると判定される。

centerCubeI, J, K を中心とする $5 \times 5 \times 5$ のボクセルについて上記の処理を行う。そして、計 125 個 (LiDAR を中心とする $5 \times 5 \times 5 = 125$ 個) のボクセルのうち、地図の内側にあるボクセルのインデックスを `_laserCloudSurroundInd` に集める。また、地図の内側にあり、かつ LiDAR の視野に入っているボクセルのインデックスを `_laserCloudValidInd` に集める。`_laserCloudSurroundInd` は `createDownsizedMap()` 内で、また `_laserCloudValidInd` は `process()` 内で利用される。

続いて、姿勢 $T_{k+1}^W = [t_{k+1}^W, R_{t+1}^W]$ の最適化の際に用いる地図点群 `_laserCloudCornerFromMap`, `_laserCloudSurfFromMap` と、最新 ($k+1$ 番目) の点群 `_laserCloudCornerStackDS`, `_laserCloudSurfStackDS` を準備する。`_laserCloudSurroundInd` が指定したインデックスをもつボクセル (点群) を `_laserCloudCornerArray` と `_laserCloudSurfArray` から取り出して、`_laserCloudCornerFromMap` と `_laserCloudSurfFromMap` に追加することで、地図点群が作成される。最適化で使用する地図点群は、LiDAR の姿勢 T_{k+1}^W の周辺に存在し、地図の内側にあつて、かつ LiDAR の視野に入っている点を集めた点群といえる。`_laserCloudCornerArray` と `_laserCloudSurfArray` は、地図内部に含まれる $21 \times 11 \times 21 = 4851$ 個のボクセル (小さな点群) のリストであり、LiDAR の周囲 $1050\text{m} \times 550\text{m} \times 1050\text{m}$ の空間にある、コーナーあるいは平面として分類された点を管理している。最適化で使用するのは地図の一部である。

`_laserCloudCornerStackDS` と `_laserCloudSurfStackDS` は、現在 ($k+1$ 番目) の点群 `_laserCloudCornerStack` と `_laserCloudSurfStack` を、ボクセルフィルタ `_downSizeFilterCorner` と `_downSizeFilterSurf` によって間引いたものである。各点の座標は、`process()` が呼ばれた時点では、時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 O_{k+2} で表されていたが、`pointAssociateToMap()` によってグローバル座標系に変換されている。ここでは更に `pointAssociateToMapped()` を呼び出すことで、グローバル座標系から、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 M_{k+2} へと変換している (図 11 参照)。最適化

で用いる点群は、コーナーと平面上の点の2種類に分けられている。従って、コーナー同士 (`_laserCloudCornerFromMap` と `_laserCloudCornerStackDS`)、あるいは平面上の点同士 (`_laserCloudSurfFromMap` と `_laserCloudSurfStackDS`) で対応付けを行う。

7.7.2 最適化による姿勢の更新

`optimizeTransformToBeMapped()` を呼び出して、最新 ($k + 1$ 番目) の点群 (`_laserCloudCornerStackDS`, `_laserCloudSurfStackDS`) と、地図点群 (`_laserCloudCornerFromMap`, `_laserCloudSurfFromMap`) との間でレジストレーションを実行し、時刻 t_{k+2} における LiDAR の姿勢 T_{k+1}^W (`_transformToBeMapped`) を最適化する。 `optimizeTransformToBeMapped()` の内部では `transformUpdate()` が呼ばれて、 `_transformAftMapped` と `_transformBefMapped` が更新される。

7.7.3 地図の更新

現在 ($k + 1$ 番目) の点群 `_laserCloudCornerStackDS` と `_laserCloudSurfStackDS` に格納されている点を地図に追加する。ここでいう地図とは、LiDAR の周囲 $1050\text{m} \times 550\text{m} \times 1050\text{m}$ の空間に含まれる点群のことであり、空間は一辺が 50m のボクセル ($21 \times 11 \times 21 = 4851$ 個) で仕切られている。 `_laserCloudCornerArray` と `_laserCloudSurfArray` は `_laserCloudNum` 個 (4851 個) のボクセルの配列であり、各ボクセルには点群が割り当てられていて、地図に含まれるコーナーと平面上の点を管理する。先ほどの最適化処理によって、時刻 t_{k+2} におけるマッピング後の LiDAR の姿勢 T_{k+1}^W が正確に求まった。この姿勢 T_{k+1}^W は、グローバル座標系から見た、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 M_{k+2} の姿勢であるから、 M_{k+2} 座標系で表される点の座標をグローバル座標系へと変換する。 `pointAssociateToMap()` を使って、各点の座標をグローバル座標系に変換し、変数 `pointSel` に格納する。点の座標 `pointSel` と対応するボクセルのインデックスを計算して、 `cubeI`, `cubeJ`, `cubeK` に代入する。ボクセルが地図の内側であれば (`cubeI`, `cubeJ`, `cubeK` が 0 から `_laserCloudWidth - 1`, 0 から `_laserCloudHeight - 1`, 0 から `_laserCloudDepth - 1` の範囲に収まっていれば)、点を地図に追加する。 `_laserCloudCornerStackDS` に含まれるコーナーであれば `_laserCloudCornerArray` に、 `_laserCloudSurfStackDS` に含まれる平面上の点であれば `_laserCloudSurfArray` に追加する。

LiDAR から見えるボクセルのインデックスの集合が `_laserCloudValidInd` に保持されている。これらのボクセルには、上記の処理によって新たな点が追加された可能性が高い。そこで、 `_laserCloudValidInd` に含まれる各インデックスについて、対応するボクセルを `_laserCloudCornerArray` と `_laserCloudSurfArray` から参照する。ボクセル内の点群を、フィルタ `_downSizeFilterCorner`, `_downSizeFilterSurf` によって間引くことで、地図 `_laserCloudCornerArray`, `_laserCloudSurfArray` に含まれる点の個数が増え過ぎないようにする。

最後に `transformFullResToMap()` を呼び出して、現在の点群 `_laserCloudFullRes` に含まれる全ての点 (M_{k+2} 座標系) をグローバル座標系に変換する。また `createDownsizedMap()` を呼び出して、LiDAR の周囲 $250\text{m} \times 250\text{m} \times 250\text{m}$ にあり、かつ地図の内側に存在するボクセル (条件を満たすインデックスの集合が `_laserCloudSurroundInd` で与えられる) を基に、LiDAR の周囲を表す点群 `_laserCloudSurround` を作成する。フィルタ `_downSizeFilterCorner` により点群 `_laserCloudSurround` がダウンサンプリングされて、 `_laserCloudSurroundDS` が得られる。 `transformFullResToMap()` により座標が変換された点群 `_laserCloudFullRes` は `/velodyne_cloud_registered` トピックから、 `createDownsizedMap()` により作成された点群 `_laserCloudSurroundDS` は `/laser_cloud_surround` トピックから Publish される。

7.8 BasicLaserMapping::optimizeTransformToBeMapped()

BasicLaserOdometry の process() と同様に、点群間の対応付けと、ガウス・ニュートン法による姿勢 T_{k+1}^W の更新を行う。このメソッドは長いので、分割して説明する。

7.8.1 コーナーの対応付け

対応付け処理は、コーナー同士 (_laserCloudCornerStackDS と _laserCloudCornerFromMap)、あるいは平面上の点同士 (_laserCloudSurfStackDS と _laserCloudSurfFromMap) で行う。_laserCloudCornerFromMap は、LiDAR の周囲にあるコーナーを集めた地図点群である。また点群 _laserCloudCornerStackDS は、現在 ($k+1$ 番目) の点群に含まれる点のうち、コーナー (それほど顕著ではないコーナー、あるいは顕著なコーナー) として分類されたものである。_laserCloudCornerFromMap を $\mathcal{M}_\mathcal{E}$ 、_laserCloudCornerStackDS を $\bar{\mathcal{A}}_{k+1}$ で表す。 $\mathcal{M}_\mathcal{E}$ と $\bar{\mathcal{A}}_{k+1}$ に含まれる各点の座標はそれぞれ、グローバル座標系、あるいは時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 (図 11 における M_{k+2}) で表される。 $\bar{\mathcal{A}}_{k+1}$ に含まれる各点 i について、その座標を M_{k+2} 座標系からグローバル座標系に変換する。点 $i \in \bar{\mathcal{A}}_{k+1}$ の元々の座標 pointOri を $X_{(k+1,i)}^L$ 、変換後のグローバル座標系での座標 pointSel を $X_{(k+1,i)}^W$ とおく。点 $X_{(k+1,i)}^W$ に対して、5 つの最近傍点を $\mathcal{M}_\mathcal{E}$ から探索する。最近傍点のセットを論文に倣って $S'_i \subset \mathcal{M}_\mathcal{E}$ とおき ($i \in \bar{\mathcal{A}}_{k+1}$)、 S'_i に含まれる各点 j の座標 (グローバル座標系) を X_j^W とする。各点 X_j^W ($j \in S'_i$) のインデックス j が pointSearchInd に、また点 X_j^W から $X_{(k+1,i)}^W$ までの二乗距離 $\|X_{(k+1,i)}^W - X_j^W\|^2$ が pointSearchSqDis に代入される。5 つの最近傍点全てが、 $X_{(k+1,i)}^W$ から 1m 以内の距離にあるなら、 $X_{(k+1,i)}^W$ と X_j^W は同じコーナーを表している、即ち対応していると考えられる。

最近傍点 X_j^W ($j \in \mathcal{M}_\mathcal{E}$) の座標の平均と共分散行列を計算し、3 次元ベクトル vc と 3×3 実対称行列 matA1 に格納する。vc と matA1 をそれぞれ $\boldsymbol{\mu}_{(k+1,i)}$ 、 $\boldsymbol{\Sigma}_{(k+1,i)}$ とすると、次の式で得られる ($|S'_i| = 5$ である)。

$$\boldsymbol{\mu}_{(k+1,i)} = \frac{1}{|S'_i|} \sum_{j \in S'_i} X_j^W \quad (122)$$

$$\boldsymbol{\Sigma}_{(k+1,i)} = \frac{1}{|S'_i|} \sum_{j \in S'_i} (X_j^W - \boldsymbol{\mu}_{(k+1,i)}) (X_j^W - \boldsymbol{\mu}_{(k+1,i)})^\top \quad (123)$$

共分散行列 $\boldsymbol{\Sigma}_{(k+1,i)}$ は対称であるから、上三角部分あるいは下三角部分のみ計算すればよい (ソースコードでは下三角部分のみ求めている)。Eigen の SelfAdjointEigenSolver を使うことで、共分散行列 $\boldsymbol{\Sigma}_{(k+1,i)}$ の固有値 $\lambda_1, \lambda_2, \lambda_3$ と固有ベクトル $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ を計算する。3 次元ベクトル matD1 は固有値をまとめたベクトル $[\lambda_1, \lambda_2, \lambda_3]^\top$ ($\lambda_1 \leq \lambda_2 \leq \lambda_3$)、 3×3 行列 matV1 は固有ベクトルをまとめた行列 $[\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]$ に対応する。共分散行列は、最近傍点 X_j^W ($j \in S'_i$) が、どの方向 (固有ベクトル) にどのくらい (固有値) 広がっているのかを表す。もし最近傍点 X_j^W がエッジ (直線) を表しているとする、直線を表す方向に点が広がっていて、他の方向には全く広がっていないはずである。従って、最大の固有値 λ_3 が他 2 つの固有値 λ_1, λ_2 も飛び抜けて大きいならば、最近傍点はエッジを表していると判断できる (ソースコード上では、 λ_3 が λ_2 の 3 倍よりも大きい場合)。このとき、最大固有値 λ_3 に対応する固有ベクトル \boldsymbol{v}_3 の方向に、直線が伸びていると考える。続いて、 $X_{(k+1,i)}^W$ と X_j^W ($j \in S'_i$) を使って、点とエッジ間の距離計算を行う。

7.8.2 点とエッジ間の距離計算

$\bar{\mathcal{A}}_{k+1}$ に含まれる点 i (グローバル座標系で $\mathbf{X}_{(k+1,i)}^W$) に対して, その最近傍点 S'_i が \mathcal{M}_ε のなかから探索された. 最近傍点はエッジを表しており, その座標 \mathbf{X}_j^W ($j \in S'_i$, グローバル座標系) の平均が $\boldsymbol{\mu}_{(k+1,i)}^W$, 共分散行列が $\boldsymbol{\Sigma}_{(k+1,i)}^W$ として求まっている. エッジの方向は, 共分散行列 $\boldsymbol{\Sigma}_{(k+1,i)}^W$ の最大固有値 λ_3 に対応する固有ベクトル \mathbf{v}_3 で与えられる. 平均 $\boldsymbol{\mu}_{(k+1,i)}^W$ は, 点 $\mathbf{X}_{(k+1,i)}^W$ に対応するエッジ上の点として捉えられる. $\boldsymbol{\mu}_{(k+1,i)}^W$ に $\pm 0.1\mathbf{v}_3$ を足した点 $\mathbf{p}_1 = \boldsymbol{\mu}_{(k+1,i)}^W + 0.1\mathbf{v}_3$, $\mathbf{p}_2 = \boldsymbol{\mu}_{(k+1,i)}^W - 0.1\mathbf{v}_3$ も, やはりエッジ上にある. 点 $\mathbf{X}_{(k+1,i)}^W$ とエッジ間の距離 d_ε は, 点 $\mathbf{X}_{(k+1,i)}^W$ と, 2 点 $\mathbf{p}_1, \mathbf{p}_2$ 間を結ぶ直線との距離として計算できる. 論文 [1] の式 (2) に倣って, 距離 d_ε は次のように書ける (BasicLaserOdometry の process() に登場した式 (20) と同様).

$$d_\varepsilon = \frac{\left\| \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) \right\|}{\left\| \mathbf{p}_1 - \mathbf{p}_2 \right\|} \quad (124)$$

各座標 $\mathbf{X}_{(k+1,i)}^W = [x_0, y_0, z_0]^\top$ は変数 x_0, y_0, z_0 , $\mathbf{p}_1 = [x_1, y_1, z_1]^\top$ は変数 x_1, y_1, z_1 , $\mathbf{p}_2 = [x_2, y_2, z_2]^\top$ は変数 x_2, y_2, z_2 に対応する. (124) 式の分子 a012 と分母 112 は, 次のように求められる.

$$\begin{aligned} \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) &= [(y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2), \\ &\quad (z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2), \\ &\quad (x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)]^\top \end{aligned} \quad (125)$$

$$\begin{aligned} \left\| \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) \right\| &= \left(((y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2))^2 \right. \\ &\quad \left. + ((z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2))^2 \right. \\ &\quad \left. + ((x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2))^2 \right)^{\frac{1}{2}} \end{aligned} \quad (126)$$

$$\left\| \mathbf{p}_1 - \mathbf{p}_2 \right\| = \left((x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \right)^{\frac{1}{2}} \quad (127)$$

a012 を 112 で割ることで (124) 式の d_ε が得られ, 1d2 に代入される. また後の最適化処理で使うために, 以下の係数ベクトル $[l_a, l_b, l_c]^\top$ (変数 $1a, 1b, 1c$ に対応) も求めておく.

$$[l_a, l_b, l_c]^\top = \frac{(\mathbf{p}_1 - \mathbf{p}_2) \times \left(\left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) \right)}{\left\| \mathbf{p}_1 - \mathbf{p}_2 \right\| \left\| \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) \right\|} \quad (128)$$

(128) 式の分母は, a012 と 112 の積として得られ, 分子は次のようになる.

$$\begin{aligned} &(\mathbf{p}_1 - \mathbf{p}_2) \times \left(\left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) \right) \\ &= [(y_1 - y_2)((x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)) \\ &\quad - (z_1 - z_2)((z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2)), \\ &\quad (z_1 - z_2)((y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2)) \\ &\quad - (x_1 - x_2)((x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)), \\ &\quad (x_1 - x_2)((z_0 - z_1)(x_0 - x_2) - (x_0 - x_1)(z_0 - z_2)) \\ &\quad - (y_1 - y_2)((y_0 - y_1)(z_0 - z_2) - (z_0 - z_1)(y_0 - y_2))]^\top \end{aligned}$$

BasicLaserOdometry の process() で説明した通り, $[l_a, l_b, l_c]^T$ はエッジの単位法線ベクトルである (エッジ上の点 p_1 と p_2 を結ぶベクトル $p_1 - p_2$ に直交する単位ベクトル). 単位法線ベクトル $[l_a, l_b, l_c]^T$ と距離 d_ε を, 変数 coeff に代入する. 誤った対応付けによって生じる外れ値に対処するため, 距離 d_ε が大きく信頼できない対応関係 ($X_{(k+1,i)}^W, X_j^W$) には 0 に近い小さな重みを, 距離 d_ε が小さく十分信頼できるものには 1 に近い重みを与える. coeff に代入する前に, 単位法線ベクトルと距離は重み s によってスケールされる. グローバル座標系に変換する前の, 元々の点 $i \in \bar{A}_{k+1}$ の座標 $X_{(k+1,i)}^L$ (M_{k+2} 座標系) を _laserCloudOri に, また coeff を _coeffSel に追加する. _laserCloudOri と _coeffSel は最適化処理で用いる.

7.8.3 平面の対応付け

平面の対応付けでは, _laserCloudSurfStackDS と _laserCloudSurfFromMap の 2 つを用いる. _laserCloudSurfFromMap は, LiDAR の周囲にある平面上の点を集めた地図点群である. また点群 _laserCloudSurfStackDS は, 現在 ($k+1$ 番目) の点群に含まれる点のうち, 平面上の点 (それほど顕著ではない平面, あるいは顕著な平面) として分類されたものである. _laserCloudSurfFromMap を \mathcal{M}_H , _laserCloudSurfStackDS を \bar{B}_{k+1} で表す. \mathcal{M}_H と \bar{B}_{k+1} に含まれる各点の座標はそれぞれ, グローバル座標系, あるいは時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 (図 11 における M_{k+2}) で表される. \bar{B}_{k+1} に含まれる各点 i について, その座標を M_{k+2} 座標系からグローバル座標系に変換する. 点 $i \in \bar{B}_{k+1}$ の元々の座標 pointOri を $X_{(k+1,i)}^L$, 変換後のグローバル座標系での座標 pointSel を $X_{(k+1,i)}^W$ とおく. 点 $X_{(k+1,i)}^W$ に対して, 5 つの最近傍点を \mathcal{M}_H から探索する. 先程と同様に, 最近傍点のセットを $S'_i \subset \mathcal{M}_H$ とおき ($i \in \bar{B}_{k+1}$), S'_i に含まれる各点 j の座標 (グローバル座標系) を X_j^W とする. 各点 X_j^W ($j \in S'_i$) のインデックス j が pointSearchInd に, また点 X_j^W から $X_{(k+1,i)}^W$ までの二乗距離 $\|X_{(k+1,i)}^W - X_j^W\|^2$ が pointSearchSqDis に代入される. 5 つの最近傍点全てが, $X_{(k+1,i)}^W$ から 1m 以内の距離にあるなら, $X_{(k+1,i)}^W$ と X_j^W は同じ平面上にあって, 対応していると考えられる. ここまでは, コーナーの対応付けと全く同じ処理である.

続いて, 同一平面上にある最近傍点 X_j^W ($j \in S'_i$) から, 平面の方程式 $[p_a, p_b, p_c]x + d = 0$ を求めて, 法線ベクトル $[p_a, p_b, p_c]^T$ (変数 pa, pb, pc に対応) を取り出す. 平面の方程式には未知数が p_a, p_b, p_c, d の 4 つあるが, これらに何らかの定数 k を掛けても, 元と同一の平面を表す ($[kp_a, kp_b, kp_c]x + kd = 0$ と $[p_a, p_b, p_c]x + d = 0$ は同じ平面である). このように, p_a, p_b, p_c, d には幾通りもの解が考えられるので, 1 つに定めるために $d = 1$ とする. 従って, 平面の方程式 $[p_a, p_b, p_c]x + 1 = 0$ をみたく法線ベクトル $[p_a, p_b, p_c]^T$ を, 最近傍点の座標 X_j^W ($j \in S'_i$) から求める. 5 つの最近傍点を X_1^W, \dots, X_5^W とおくと, 以下の 5 つの方程式が得られる.

$$\begin{bmatrix} X_1^W & X_2^W & X_3^W & X_4^W & X_5^W \end{bmatrix}^T \begin{bmatrix} p_a \\ p_b \\ p_c \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad (129)$$

未知数が 3 つであるのに対して制約 (方程式) が 5 つもあるから, 5 つの制約を可能な限り満たすように, 最小二乗法を使って解を計算する. 上式の左辺の 5×3 行列を A , 右辺の 5 次ベクトルを b とすると, 最小二乗法による解は次のようにして得られる (行列 $(A^T A)^{-1} A^T$ は A の疑似逆行列である).

$$[p_a, p_b, p_c]^T = (A^T A)^{-1} A^T b \quad (130)$$

このような解は Eigen の ColPivHouseholderQR を使えば得られる. 法線ベクトル $[p_a, p_b, p_c]^T$ を, そのノルム $p_s = (p_a^2 + p_b^2 + p_c^2)^{\frac{1}{2}}$ (変数 ps) で割ることで, 単位法線ベクトル $[p_a/p_s, p_b/p_s, p_c/p_s]^T$ を得る. 法線ベク

トルは、平面の方程式 $[p_a, p_b, p_c] \mathbf{x} + 1 = 0$ を可能な限り満たすように求めた。従って、計算に使用した最近傍点の座標 \mathbf{X}_j^W をこの方程式、あるいは同じ意味をもつ方程式 $[p_a/p_s, p_b/p_s, p_c/p_s] \mathbf{x} + 1/p_s = 0$ に代入すれば、0に近い値が得られるはずである。言い換えれば、 $[p_a/p_s, p_b/p_s, p_c/p_s] \mathbf{X}_j^W + d/p_s \simeq 0$ が、全ての $j \in \mathcal{S}'_i$ に対して期待される。値が0から大きく外れるなら、5つの最近傍点 \mathbf{X}_j^W は同一の平面を表していない可能性が高く、 $\mathbf{X}_{(k+1,i)}^W$ と \mathbf{X}_j^W との対応付けは誤りだと考えられる。ソースコード内では、5つの最近傍点の座標 $\mathbf{X}_1^W, \dots, \mathbf{X}_5^W$ を順番に方程式に代入していき、計算結果が -0.2 から 0.2 の範囲に収まらなければ、平面ではないと判断する (planeValid を false にセットする)。そのような誤った対応関係は、後続の最適化処理では使われない。

7.8.4 点と平面間の距離計算

点 $i (i \in \bar{\mathcal{B}}_{k+1})$ に対応する最近傍点の集合 $\mathcal{S}'_i \in \mathcal{M}_{\mathcal{H}}$ が、同一平面であることを確認できたら、この平面と点 $i (\mathbf{X}_{(k+1,i)}^W)$ との距離 $d_{\mathcal{H}}$ を次のように計算する ($p_s = (p_a^2 + p_b^2 + p_c^2)^{\frac{1}{2}}$)。距離 $d_{\mathcal{H}}$ は変数 pd2 に代入される。

$$d_{\mathcal{H}} = \frac{[p_a, p_b, p_c] \mathbf{X}_{(k+1,i)}^W + p_d}{(p_a^2 + p_b^2 + p_c^2)^{\frac{1}{2}}} = \left[\frac{p_a}{p_s}, \frac{p_b}{p_s}, \frac{p_c}{p_s} \right] \mathbf{X}_{(k+1,i)}^W + \frac{p_d}{p_s} \quad (131)$$

先程と同様に、距離 $d_{\mathcal{H}}$ が大きく信頼できない対応関係 $\mathbf{X}_{(k+1,i)}^W, \mathbf{X}_j^W$ には、0に近い小さな重みを与える。単位法線ベクトル $[p_a/p_s, p_b/p_s, p_c/p_s]^{\top}$ と距離 $d_{\mathcal{H}}$ は、重み係数 (変数 s) によってスケールされ、変数 coeff に代入される。その後、グローバル座標系に変換する前の、元々の点 $i \in \bar{\mathcal{B}}_{k+1}$ の座標 $\mathbf{X}_{(k+1,i)}^L (M_{k+2}$ 座標系) を _laserCloudOri に、また coeff を _coeffSel に追加する。

7.8.5 最適化による姿勢の更新

上記の対応付け処理の後には、最適化処理、具体的にはガウス・ニュートン法による姿勢 \mathbf{T}_{k+1}^W (_transform-TobeMapped) の更新が控えている。基本的には、BasicLaserOdometry の process() での最適化処理と同様である。(124) 式に示す、点とエッジ間の距離 $d_{\mathcal{E}}$ と、(131) 式に示す、点と平面間の距離 $d_{\mathcal{H}}$ を最小化するような姿勢 \mathbf{T}_{k+1}^W が逐次的に計算される。 $\mathbf{T}_{k+1}^W = [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}, \theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}]^{\top}$ は時刻 t_{k+2} におけるマッピング後の LiDAR の姿勢、即ちグローバル座標系からみた、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 M_{k+2} の相対姿勢 (図 11) であり、 M_{k+2} 座標系で表される点の座標 \mathbf{p}^L を、グローバル座標系 \mathbf{p}^G へと変換する。

$$\begin{aligned} \mathbf{p}^G &= \mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}) \mathbf{p}^L + [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^{\top} \\ &= \mathbf{R}_y(\theta_y^{k+1,W}) \mathbf{R}_x(\theta_x^{k+1,W}) \mathbf{R}_z(\theta_z^{k+1,W}) \mathbf{p}^L + [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^{\top} \end{aligned} \quad (132)$$

上式を (124) 式と (131) 式に代入することで、距離 $d_{\mathcal{E}}$ と $d_{\mathcal{H}}$ を \mathbf{T}_{k+1}^W の関数 $f_{\mathcal{E}}, f_{\mathcal{H}}$ として書き直せる。 $f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ における $\mathbf{X}_{(k+1,i)}^L$ は、点群 $\bar{\mathcal{A}}_{k+1}$ に属する点 i の座標である。また、 $f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ における $\mathbf{X}_{(k+1,i)}^L$ は、点群 $\bar{\mathcal{B}}_{k+1}$ に属する点 i の座標である。

$$\begin{aligned} f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) &= \frac{\|(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1) \times (\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2)\|}{\|\mathbf{p}_1 - \mathbf{p}_2\|} \\ &= \|\mathbf{p}_1 - \mathbf{p}_2\|^{-1} \\ &\quad \left\| \left(\mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}) \mathbf{p}^L + [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^{\top} - \mathbf{p}_1 \right) \right\| \end{aligned}$$

$$\begin{aligned}
& \times \left(\mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}) \mathbf{p}^L + [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^\top - \mathbf{p}_2 \right) \Big\| \quad (133) \\
f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) &= \begin{bmatrix} p_a & p_b & p_c \\ p_s & p_s & p_s \end{bmatrix} \mathbf{X}_{(k+1,i)}^W + \frac{p_d}{p_s} \\
&= \begin{bmatrix} p_a & p_b & p_c \\ p_s & p_s & p_s \end{bmatrix} (\mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}) \mathbf{p}^L \\
&\quad + [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^\top) + \frac{p_d}{p_s} \quad (134)
\end{aligned}$$

ガウス・ニュートン法を使って、 $d_{\mathcal{E}} = f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ と $d_{\mathcal{H}} = f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ を \mathbf{T}_{k+1}^W に関して最小化したいが、そのためには $f_{\mathcal{E}}$ と $f_{\mathcal{H}}$ の、 \mathbf{T}_{k+1}^W に関する勾配ベクトル (微分) $\nabla f_{\mathcal{E}}$, $\nabla f_{\mathcal{H}}$ が必要である。スカラー関数 $f_{\mathcal{E}}$, $f_{\mathcal{H}}$ の 6 次元ベクトル \mathbf{T}_{k+1}^W による微分だから、 $\nabla f_{\mathcal{E}}$, $\nabla f_{\mathcal{H}}$ も 6 次元ベクトルとなる。BasicLaser0dometry の process() のときと同じように、微分に関する以下の関係式を用いる。

$$\frac{d}{dx} \|(\mathbf{v}(x) - \mathbf{v}_1) \times (\mathbf{v}(x) - \mathbf{v}_2)\|^2 = 2 \frac{d}{dx} \mathbf{v}(x)^\top ((\mathbf{v}_1 - \mathbf{v}_2) \times ((\mathbf{v}(x) - \mathbf{v}_1) \times (\mathbf{v}(x) - \mathbf{v}_2))) \quad (135)$$

$$\frac{d}{dy} \|\mathbf{v}(y)\| = \frac{1}{2} \|\mathbf{v}(y)\|^{-1} \frac{d}{dy} \|\mathbf{v}(y)\|^2 \quad (136)$$

関数 $f_{\mathcal{E}}$ の勾配ベクトル $\nabla f_{\mathcal{E}}$ は、 \mathbf{T}_{k+1}^W の各要素 $t_x^{k+1,W}$, $t_y^{k+1,W}$, $t_z^{k+1,W}$, $\theta_x^{k+1,W}$, $\theta_y^{k+1,W}$, $\theta_z^{k+1,W}$ に関する、 $f_{\mathcal{E}}$ の偏微分を並べたものである。

$$\nabla f_{\mathcal{E}} = \left[\frac{\partial f_{\mathcal{E}}}{\partial t_x^{k+1,W}}, \frac{\partial f_{\mathcal{E}}}{\partial t_y^{k+1,W}}, \frac{\partial f_{\mathcal{E}}}{\partial t_z^{k+1,W}}, \frac{\partial f_{\mathcal{E}}}{\partial \theta_x^{k+1,W}}, \frac{\partial f_{\mathcal{E}}}{\partial \theta_y^{k+1,W}}, \frac{\partial f_{\mathcal{E}}}{\partial \theta_z^{k+1,W}} \right]^\top \quad (137)$$

以下では簡単のため $\mathbf{R} = \mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W})$, $\mathbf{t} = [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^\top$ とおく。 $t_x^{k+1,W}$ に関する $f_{\mathcal{E}}$ の偏微分は次のようになる。

$$\begin{aligned}
& \frac{\partial}{\partial t_x^{k+1,W}} f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \\
&= \frac{\partial}{\partial t_x^{k+1,W}} \|\mathbf{p}_1 - \mathbf{p}_2\|^{-1} \left\| \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right\| \\
&= \|\mathbf{p}_1 - \mathbf{p}_2\|^{-1} \frac{\partial}{\partial t_x^{k+1,W}} \left\| \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right\| \\
&= \frac{1}{2} \|\mathbf{p}_1 - \mathbf{p}_2\|^{-1} \left\| \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right\|^{-1} \\
&\quad \frac{\partial}{\partial t_x^{k+1,W}} \left(\left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right) \\
&= \frac{1}{2} \|\mathbf{p}_1 - \mathbf{p}_2\|^{-1} \left\| \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right\|^{-1} \\
&\quad 2 \left(\frac{\partial}{\partial t_x^{k+1,W}} \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) \right)^\top \\
&\quad \left(\mathbf{p}_1 - \mathbf{p}_2 \right) \times \left(\left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right) \\
&= \|\mathbf{p}_1 - \mathbf{p}_2\|^{-1} \left\| \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right\|^{-1} \left(\frac{\partial}{\partial t_x^{k+1,W}} \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) \right)^\top \\
&\quad \left(\left(\mathbf{p}_1 - \mathbf{p}_2 \right) \times \left(\left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_1 \right) \times \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} - \mathbf{p}_2 \right) \right) \right) \\
&= \|\mathbf{p}_1 - \mathbf{p}_2\|^{-1} \left\| \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) \right\|^{-1} \left(\frac{\partial}{\partial t_x^{k+1,W}} \left(\mathbf{R} \mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) \right)^\top
\end{aligned}$$

$$\left((\mathbf{p}_1 - \mathbf{p}_2) \times \left(\left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_1 \right) \times \left(\mathbf{X}_{(k+1,i)}^W - \mathbf{p}_2 \right) \right) \right) \quad (138)$$

最後の式変形では $\mathbf{X}_{(k+1,i)}^W = \mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t}$ を再度使っている。(128) 式の単位法線ベクトル $[l_a, l_b, l_c]^\top$ を使えば以下を得る。

$$\frac{\partial}{\partial t_x^{k+1,W}} f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = \left(\frac{\partial}{\partial t_x^{k+1,W}} \left(\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) \right)^\top [l_a, l_b, l_c]^\top \quad (139)$$

$t_y^{k+1,W}, t_z^{k+1,W}, \theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}$ に関する偏微分も、上と同様の式になる ($\partial t_x^{k+1,W}$ の部分だけを置き換えればよい) から、あとは $\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t}$ の各変数に関する偏微分が分かればよい。続いて、関数 $f_{\mathcal{H}}$ を \mathbf{T}_{k+1}^W の各要素について偏微分し、以下に示す勾配ベクトル $\nabla f_{\mathcal{H}}$ を求める。

$$\nabla f_{\mathcal{H}} = \left[\frac{\partial f_{\mathcal{H}}}{\partial t_x^{k+1,W}}, \frac{\partial f_{\mathcal{H}}}{\partial t_y^{k+1,W}}, \frac{\partial f_{\mathcal{H}}}{\partial t_z^{k+1,W}}, \frac{\partial f_{\mathcal{H}}}{\partial \theta_x^{k+1,W}}, \frac{\partial f_{\mathcal{H}}}{\partial \theta_y^{k+1,W}}, \frac{\partial f_{\mathcal{H}}}{\partial \theta_z^{k+1,W}} \right]^\top \quad (140)$$

先程と同様に $\mathbf{R} = \mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W})$, $\mathbf{t} = [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}]^\top$ とおく。 $t_x^{k+1,W}$ に関する $f_{\mathcal{H}}$ の偏微分は次のようになる。

$$\begin{aligned} \frac{\partial}{\partial t_x^{k+1,W}} f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) &= \frac{\partial}{\partial t_x^{k+1,W}} \left(\left[\frac{p_a}{p_s}, \frac{p_b}{p_s}, \frac{p_c}{p_s} \right] \left(\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) + \frac{p_d}{p_s} \right) \\ &= \left[\frac{p_a}{p_s}, \frac{p_b}{p_s}, \frac{p_c}{p_s} \right] \left(\frac{\partial}{\partial t_x^{k+1,W}} \left(\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) \right) \\ &= \left(\frac{\partial}{\partial t_x^{k+1,W}} \left(\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) \right)^\top \left[\frac{p_a}{p_s}, \frac{p_b}{p_s}, \frac{p_c}{p_s} \right]^\top \end{aligned} \quad (141)$$

$t_y^{k+1,W}, t_z^{k+1,W}, \theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}$ に関する偏微分も、上と同様であるから、あとは $\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t}$ の各変数に関する偏微分が分かればよい。(139) 式と (141) 式は同じ形をしているので、関数 $f_{\mathcal{E}}, f_{\mathcal{H}}$ の偏微分の計算は、1 つのソースコードに共通化できる。回転行列 $\mathbf{R} = \mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W})$ の各要素は次のように表される。

$$\begin{aligned} &\mathbf{R}(\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}) \\ &= \mathbf{R}_y(\theta_y^{k+1,W}) \mathbf{R}_x(\theta_x^{k+1,W}) \mathbf{R}_z(\theta_z^{k+1,W}) \\ &= \begin{bmatrix} \cos \theta_y^{k+1,W} \cos \theta_z^{k+1,W} + \sin \theta_y^{k+1,W} \sin \theta_x^{k+1,W} \sin \theta_z^{k+1,W} & & \\ & \cos \theta_x^{k+1,W} \sin \theta_z^{k+1,W} & \\ -\sin \theta_y^{k+1,W} \cos \theta_z^{k+1,W} + \cos \theta_y^{k+1,W} \sin \theta_x^{k+1,W} \sin \theta_z^{k+1,W} & & \\ & -\cos \theta_x^{k+1,W} \cos \theta_z^{k+1,W} & \\ \sin \theta_y^{k+1,W} \sin \theta_z^{k+1,W} + \cos \theta_y^{k+1,W} \sin \theta_x^{k+1,W} \cos \theta_z^{k+1,W} & \sin \theta_y^{k+1,W} \cos \theta_x^{k+1,W} & \\ & -\sin \theta_x^{k+1,W} & \\ & \cos \theta_y^{k+1,W} \cos \theta_x^{k+1,W} & \end{bmatrix} \quad (142) \end{aligned}$$

$\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t}$ の $t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}$ に関する偏微分は次の通りである。

$$\frac{\partial}{\partial t_x^{k+1,W}} \left(\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) = [1, 0, 0]^\top \quad (143)$$

$$\frac{\partial}{\partial t_y^{k+1,W}} \left(\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) = [0, 1, 0]^\top \quad (144)$$

$$\frac{\partial}{\partial t_z^{k+1,W}} \left(\mathbf{R}\mathbf{X}_{(k+1,i)}^L + \mathbf{t} \right) = [0, 0, 1]^\top \quad (145)$$

$$\frac{\partial}{\partial \theta_y^{k+1,W}} f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_y^{k+1,W}} \mathbf{R} \right) \mathbf{X}_{(k+1,i)}^L \right)^{\top} [l_a, l_b, l_c]^{\top} \quad (153)$$

$$\frac{\partial}{\partial \theta_z^{k+1,W}} f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_z^{k+1,W}} \mathbf{R} \right) \mathbf{X}_{(k+1,i)}^L \right)^{\top} [l_a, l_b, l_c]^{\top} \quad (154)$$

$$\frac{\partial}{\partial t_x^{k+1,W}} f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = [1, 0, 0] \begin{bmatrix} p_a/p_s & p_b/p_s & p_c/p_s \end{bmatrix}^{\top} = \frac{p_a}{p_s} \quad (155)$$

$$\frac{\partial}{\partial t_y^{k+1,W}} f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = [0, 1, 0] \begin{bmatrix} p_a/p_s & p_b/p_s & p_c/p_s \end{bmatrix}^{\top} = \frac{p_b}{p_s} \quad (156)$$

$$\frac{\partial}{\partial t_z^{k+1,W}} f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = [0, 0, 1] \begin{bmatrix} p_a/p_s & p_b/p_s & p_c/p_s \end{bmatrix}^{\top} = \frac{p_c}{p_s} \quad (157)$$

$$\frac{\partial}{\partial \theta_x^{k+1,W}} f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_x^{k+1,W}} \mathbf{R} \right) \mathbf{X}_{(k+1,i)}^L \right)^{\top} \begin{bmatrix} p_a/p_s & p_b/p_s & p_c/p_s \end{bmatrix}^{\top} \quad (158)$$

$$\frac{\partial}{\partial \theta_y^{k+1,W}} f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_y^{k+1,W}} \mathbf{R} \right) \mathbf{X}_{(k+1,i)}^L \right)^{\top} \begin{bmatrix} p_a/p_s & p_b/p_s & p_c/p_s \end{bmatrix}^{\top} \quad (159)$$

$$\frac{\partial}{\partial \theta_z^{k+1,W}} f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = \left(\left(\frac{\partial}{\partial \theta_z^{k+1,W}} \mathbf{R} \right) \mathbf{X}_{(k+1,i)}^L \right)^{\top} \begin{bmatrix} p_a/p_s & p_b/p_s & p_c/p_s \end{bmatrix}^{\top} \quad (160)$$

ソースコード内の変数とは次のように対応する。

$$\begin{aligned} \text{srx} &= \sin(\theta_x^{k+1,W}), & \text{sry} &= \sin(\theta_y^{k+1,W}), & \text{srz} &= \sin(\theta_z^{k+1,W}) \\ \text{crx} &= \cos(\theta_x^{k+1,W}), & \text{cry} &= \cos(\theta_y^{k+1,W}), & \text{crz} &= \cos(\theta_z^{k+1,W}) \\ \text{tx} &= t_x^{k+1,W}, & \text{ty} &= t_y^{k+1,W}, & \text{tz} &= t_z^{k+1,W} \end{aligned}$$

$$\begin{aligned} \text{arx} &= \frac{\partial f_{\mathcal{E}}}{\partial \theta_x^{k+1,W}} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial \theta_x^{k+1,W}}, & \text{ary} &= \frac{\partial f_{\mathcal{E}}}{\partial \theta_y^{k+1,W}} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial \theta_y^{k+1,W}}, & \text{arz} &= \frac{\partial f_{\mathcal{E}}}{\partial \theta_z^{k+1,W}} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial \theta_z^{k+1,W}} \\ \text{atx} &= \frac{\partial f_{\mathcal{E}}}{\partial t_x^{k+1,W}} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial t_x^{k+1,W}}, & \text{aty} &= \frac{\partial f_{\mathcal{E}}}{\partial t_y^{k+1,W}} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial t_y^{k+1,W}}, & \text{atz} &= \frac{\partial f_{\mathcal{E}}}{\partial t_z^{k+1,W}} \text{ または } \frac{\partial f_{\mathcal{H}}}{\partial t_z^{k+1,W}} \end{aligned}$$

係数ベクトル $[l_a, l_b, l_c]^{\top}$ および $[p_a/p_s, p_b/p_s, p_c/p_s]^{\top}$ は `_coeffSel` から取り出せる。点 i ($i \in \bar{\mathcal{A}}_{k+1}$ あるいは $i \in \bar{\mathcal{B}}_{k+1}$) の座標 $\mathbf{X}_{(k+1,i)}^L$ も、`_laserCloudOri` から得られる (M_{k+2} 座標系)。従って、偏微分の計算に必要な要素は全て揃っている。各点 i について勾配ベクトル $\nabla f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$, $\nabla f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ が求まったら、ガウス・ニュートン法を使って姿勢 \mathbf{T}_{k+1}^W を更新する。

姿勢を更新するために最小化する関数は、以下に示す通り、距離 (残差) の二乗和である。

$$\mathcal{L}(\mathbf{T}_{k+1}^W) = \sum_{i \in \bar{\mathcal{A}}_{k+1}} \left(f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \right)^2 + \sum_{i \in \bar{\mathcal{B}}_{k+1}} \left(f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \right)^2 \quad (161)$$

$$(\mathbf{T}_{k+1}^W)^* = \arg \min_{\mathbf{T}_{k+1}^W} \mathcal{L}(\mathbf{T}_{k+1}^W) \quad (162)$$

残差 $d_{\mathcal{E}} = f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$, $d_{\mathcal{H}} = f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ は \mathbf{T}_{k+1}^W についての関数である。これを、初期値 $\check{\mathbf{T}}_{k+1}^W$ からの変動 $\Delta \mathbf{T}_{k+1}^W$ の関数として書き直す ($\mathbf{T}_{k+1}^W = \check{\mathbf{T}}_{k+1}^W + \Delta \mathbf{T}_{k+1}^W$)。

$$f_{\mathcal{E}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W + \Delta \mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad (163)$$

$$f_{\mathcal{H}}(\mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) = f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W + \Delta \mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad (164)$$

上記を1次の項までテイラー展開し、初期値 $\check{\mathbf{T}}_{k+1}^W$ のまわりで線形近似する。

$$f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W + \Delta \mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \simeq f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^W \quad (165)$$

$$f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W + \Delta \mathbf{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \simeq f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^W \quad (166)$$

これらを目的関数に代入して、初期値からの変動 $\Delta \mathbf{T}_{k+1}^W$ の関数として書き直す。

$$\begin{aligned} \mathcal{L}(\Delta \mathbf{T}_{k+1}^W) &= \sum_{i \in \bar{\mathcal{A}}_{k+1}} \left(f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^W \right)^2 \\ &\quad + \sum_{i \in \bar{\mathcal{B}}_{k+1}} \left(f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^W \right)^2 \end{aligned} \quad (167)$$

$\Delta \mathbf{T}_{k+1}^W$ に関する偏微分は、以下ようになる。

$$\begin{aligned} &\frac{\partial}{\partial \Delta \mathbf{T}_{k+1}^W} \mathcal{L}(\Delta \mathbf{T}_{k+1}^W) \\ &= 2 \sum_{i \in \bar{\mathcal{A}}_{k+1}} \left(f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^W \right) \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \\ &\quad + 2 \sum_{i \in \bar{\mathcal{B}}_{k+1}} \left(f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) + \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \Delta \mathbf{T}_{k+1}^W \right) \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \end{aligned} \quad (168)$$

これを0とおいて $\Delta \mathbf{T}_{k+1}^W$ について解けば、 $\mathcal{L}(\Delta \mathbf{T}_{k+1}^W)$ を最小化する最適な更新量 $(\Delta \mathbf{T}_{k+1}^W)^*$ が得られる。

$$\begin{aligned} &\left(\sum_{i \in \bar{\mathcal{A}}_{k+1}} \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \right. \\ &\quad \left. + \sum_{i \in \bar{\mathcal{B}}_{k+1}} \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)^\top \right) \Delta \mathbf{T}_{k+1}^W \\ &= - \left(\sum_{i \in \bar{\mathcal{A}}_{k+1}} \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \right. \\ &\quad \left. + \sum_{i \in \bar{\mathcal{B}}_{k+1}} \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \right) \end{aligned} \quad (169)$$

行列 $\mathbf{J}_{\mathcal{E}}, \mathbf{J}_{\mathcal{H}}$, ベクトル $d_{\mathcal{E}}, d_{\mathcal{H}}$ を以下のように定めれば ($i \in \bar{\mathcal{A}}_{k+1}$ または $i \in \bar{\mathcal{B}}_{k+1}$)

$$\mathbf{J}_{\mathcal{E}} = \left[\nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,0)}^L) \quad \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (170)$$

$$\mathbf{J}_{\mathcal{H}} = \left[\nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,0)}^L) \quad \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (171)$$

$$d_{\mathcal{E}} = \left[f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,0)}^L) \quad f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (172)$$

$$d_{\mathcal{H}} = \left[f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,0)}^L) \quad f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,1)}^L) \quad \cdots \quad f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (173)$$

更に、行列 \mathbf{J} とベクトル d を次のように定めれば

$$\mathbf{J} = \left[\mathbf{J}_{\mathcal{E}}^\top \quad \mathbf{J}_{\mathcal{H}}^\top \right]^\top = \left[\cdots \quad \nabla f_{\mathcal{E}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \quad \nabla f_{\mathcal{H}}(\check{\mathbf{T}}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) \quad \cdots \right]^\top \quad (174)$$

$$d = \begin{bmatrix} d_{\mathcal{E}}^{\top} & d_{\mathcal{H}}^{\top} \end{bmatrix}^{\top} = \begin{bmatrix} \cdots & f_{\mathcal{E}}(\check{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) & \cdots & f_{\mathcal{H}}(\check{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L) & \cdots \end{bmatrix}^{\top} \quad (175)$$

最適な ΔT_{k+1}^W は次の式から得られる.

$$(\mathbf{J}^{\top} \mathbf{J}) \Delta T_{k+1}^W = -\mathbf{J}^{\top} d \quad (176)$$

$$\Rightarrow (\Delta T_{k+1}^W)^* = -(\mathbf{J}^{\top} \mathbf{J})^{-1} \mathbf{J}^{\top} d \quad (177)$$

この更新量 ΔT_{k+1}^W を初期値 \check{T}_{k+1}^W (現在の_transformTobeMapped) に加えることで、姿勢の解 T_{k+1}^W を更新する. 行列とベクトルのサイズを整理しておくと、ヤコビ行列 $J_{\mathcal{E}}$ は $|\bar{A}_{k+1}| \times 6$, $J_{\mathcal{H}}$ は $|\bar{B}_{k+1}| \times 6$, J は $(|\bar{A}_{k+1}| + |\bar{B}_{k+1}|) \times 6$, 残差ベクトル $d_{\mathcal{E}}$ は $|\bar{A}_{k+1}|$, $d_{\mathcal{H}}$ は $|\bar{B}_{k+1}|$, d は $|\bar{A}_{k+1}| + |\bar{B}_{k+1}|$ である. 但し $|\bar{A}_{k+1}|$ と $|\bar{B}_{k+1}|$ は、点群 \bar{A}_{k+1} と \bar{B}_{k+1} に含まれる点の個数を表す (最新の点群のなかで、コーナーおよび平面に分類された点の個数).

ヤコビ行列 J と残差ベクトル d は、変数 matA, matB に対応する. 各点 $i \in \bar{A}_{k+1}$, $i \in \bar{B}_{k+1}$ に対して、現在の姿勢 \check{T}_{k+1}^W (_transformTobeMapped) のもとで勾配ベクトル $\nabla f_{\mathcal{E}}(\check{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$, $\nabla f_{\mathcal{H}}(\check{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ を計算する. 勾配の各要素 (arx, ary, arz, coeff.x, coeff.y, coeff.z) を、 J の点 i に対応する行 (matA(i, 0) から matA(i, 5)) に格納する. 先ほどみた通り、並進成分に関する勾配は、単に係数ベクトル $[l_a, l_b, l_c]^{\top}$, $[p_a/p_s, p_b/p_s, p_c/p_s]^{\top}$ (変数 coeff) の各要素となるから、新たな計算は不要である. また、点 i に対応する距離 (残差) $d_{\mathcal{E}} = f_{\mathcal{E}}(\check{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$, $d_{\mathcal{H}} = f_{\mathcal{H}}(\check{T}_{k+1}^W, \mathbf{X}_{(k+1,i)}^L)$ も既に計算されているが、これを d の点 i に対応する要素 matB(i, 0) に格納する. 但し、符号は反転されているので、matB が表すベクトルは $-d$ である. matAt は J^{\top} , matAtA は $J^{\top} J$, matAtB は $-\mathbf{J}^{\top} d$ に対応する. (177) 式の結果、即ち $J^{\top} J \Delta T_{k+1}^W = -\mathbf{J}^{\top} d$ を満たす最適な更新量 ΔT_{k+1}^W が matX に代入される.

ヘッセ行列 $J^{\top} J$ の固有値と固有ベクトルを手掛かりに縮退の有無を確認し、もし縮退が発生していれば更新量 ΔT_{k+1}^W を修正する. 修正のための手法は、BasicLaserOdometry の process() に記述した通りである (実装の誤りはここでもみられる). 縮退かどうかの判別に用いる固有値の閾値が、BasicLaserOdometry と比べて 10 倍に設定されているから、縮退の発生がより厳しくチェックされる.

以上で、ガウス・ニュートン法の 1 回のイテレーションにおける処理が終わって、初期値_transformTobeMapped(T_{k+1}^W) からの更新量 matX(ΔT_{k+1}^W) が得られた. _transformTobeMapped に matX の各要素を足し合わせて ($T_{k+1}^W \leftarrow T_{k+1}^W + \Delta T_{k+1}^W$), 現在の姿勢を更新する. 更新量 ΔT_{k+1}^W の並進成分と回転成分のノルム deltaT, deltaR を計算し、双方が閾値_deltaTAbort, _deltaTAbort 未満であれば、解は収束したものとみなして、ガウス・ニュートン法を終了し、現在の T_{k+1}^W を最終的な解として返す. そうでなければ、新しい解のもとで対応付けと距離計算を行って、再度解を更新する (解 T_{k+1}^W の更新に伴って、点 $i \in \bar{A}_{k+1}$, $i \in \bar{B}_{k+1}$ のグローバル座標 $X_{(k+1,i)}^W$ が変化するため、点群間の対応付けからやり直す必要がある).

最後の仕上げとして transformUpdate() を呼び出し、姿勢 T_{k+1}^W に IMU データを融合させる. その後、_transformTobeMapped を _transformAftMapped, _transformSum を _transformBefMapped に代入する. _transformAftMapped は T_k^W から T_{k+1}^W , _transformBefMapped は T_k^S から T_{k+1}^S へと更新され、transformAssociateMap() によって次の姿勢 T_{k+2}^W の初期値を計算するために用いられる.

8 LaserMapping クラス

8.1 LaserMapping::setup()

ノードの設定を ROS から受け取って、次のパラメータを設定する.

- `_scanPeriod(BasicLaserOdometry)`: 点群の取得間隔 (デフォルトでは 100ms).
- `_maxIterations(BasicLaserOdometry)`: ガウス・ニュートン法の最大のイテレーション数 (デフォルトでは 10).
- `_deltaTAbort(BasicLaserOdometry)`: ガウス・ニュートン法の解 (姿勢) の収束判定に用いる, 並進成分の変化量の閾値 (デフォルトでは 0.05).
- `_deltaRAbort(BasicLaserOdometry)`: ガウス・ニュートン法の解 (姿勢) の収束判定に用いる, 回転成分の変化量の閾値 (デフォルトでは 0.05).
- `cornerFilterSize(BasicLaserOdometry)`: `LaserOdometry` ノードの `/laser_cloud_corner_last` トピックから受け取った $k+1$ 番目の点群は, それほど顕著ではないコーナー, あるいは顕著なコーナーとして分類された点を含んでいる. この点群を間引くために適用するボクセルフィルタ (`_downSizeFilterCorner`) のサイズを決定する (デフォルトでは 0.2m).
- `surfaceFilterSize(BasicLaserOdometry)`: `LaserOdometry` ノードの `/laser_cloud_surf_last` トピックから受け取った $k+1$ 番目の点群は, それほど顕著ではない平面, あるいは平面として分類された点を含んでいる. この点群を間引くために適用するボクセルフィルタ (`_downSizeFilterSurf`) のサイズを決定する (デフォルトでは 0.4m).
- `mapFilterSize(BasicLaserOdometry)`: `LaserOdometry` ノードの `/velodyne_cloud_3` トピックから受け取った $k+1$ 番目の点群を間引くために適用する, ボクセルフィルタ (`_downSizeFilterMap`) のサイズを決定する. このボクセルフィルタは未使用である.

続いて, ノードが Subscribe あるいは Publish するトピックを設定する. Subscribe するのは次の 5 つのトピックである. `/laser_cloud_corner_last`, `/laser_cloud_surf_last`, `/velodyne_cloud_3` の 3 つのトピックは, `LaserOdometry` ノードから 5Hz の頻度で送出される (デフォルトのパラメータ設定の場合).

- `/laser_cloud_corner_last` トピック (`sensor_msgs::PointCloud2` 型): それほど顕著ではないコーナー, あるいは顕著なコーナーとして分類された点群 ($k+1$ 番目). 各点の座標は, 時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 (`/laser_odom` フレーム) に揃えられている. `BasicLaserOdometry::process()` のコーナーの対応付け処理で説明した, 点群 \bar{A}_{k+1} に相当する.
- `/laser_cloud_surf_last` トピック (`sensor_msgs::PointCloud2` 型): それほど顕著ではない平面, あるいは顕著な平面として分類された点群 ($k+1$ 番目). 各点の座標は, 時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 (`/laser_odom` フレーム) に揃えられている. `BasicLaserOdometry::process()` の平面の対応付け処理で説明した, 点群 \bar{B}_{k+1} に相当する.
- `/velodyne_cloud_3` トピック (`sensor_msgs::PointCloud2` 型): $k+1$ 番目の点群. 各点の座標は, 時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 (`/laser_odom` フレーム) に揃えられている.
- `/laser_odom_to_init` トピック (`nav_msgs::Odometry` 型): オドメトリによって得られた, 時刻 t_{k+2} における LiDAR の姿勢 (グローバル座標系). 言い換えると, グローバル座標系 (`camera_init` フレーム) からみた, 時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 (`/laser_odom` フレーム) の姿勢である. `BasicLaserOdometry::process()` の最適化処理で説明した `_transformSum`, 即ち $T_{k+1}^S = [t_x^{k+1}, t_y^{k+1}, t_z^{k+1}, \theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}]^T$ に対応する.
- `/imu/data` トピック (`sensor_msgs::Imu` 型): IMU データ (速度と加速度). マッピングによって得られた, 時刻 t_{k+2} における LiDAR の姿勢 (グローバル座標系) `_transformAftMapped` を補正するために利用される. 言い換えると, グローバル座標系 (`camera_init` フレーム) からみた, 時刻 t_{k+2} における

マッピング後の LiDAR 座標系 (/aft_mapped フレーム) の姿勢が `_transformAftMapped` である。この `_transformAftMapped` は、論文 [1] の $T_{k+1}^W = [t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}, \theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}]^T$ に相当する。

Publish するのは次の 3 つのトピックである。点群データは 5Hz の頻度で到着するので、以下に示す 3 つのトピックは 5Hz の頻度で Publish される。

- /laser_cloud_surround トピック (`sensor_msgs::PointCloud2` 型): 現在の LiDAR の位置を中心とする、周囲 2.5m の領域に含まれる点群。各点の座標はグローバル座標系 (`camera_init`) で表される。
- /velodyne_cloud_registered トピック (`velodyne_cloud_registered` 型): $k + 1$ 番目の点群。各点の座標はグローバル座標系 (`camera_init`) で表される。
- /aft_mapped_to_init トピック (`nav_msgs::Odometry` 型): マッピングあるいはオドメトリによって得られた、時刻 t_{k+2} における LiDAR の姿勢 (グローバル座標系)。言い換えると、グローバル座標系 (`camera_init` フレーム) からみた、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 (/aft_mapped フレーム)、あるいは時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 (/laser_odom フレーム) の姿勢である。この姿勢は `BasicLaserMapping` の `_transformAftMapped` と `_transformBefMapped` メンバに格納されており、それぞれ T_{k+1}^W と T_{k+1}^S に対応する。

8.2 LaserMapping::laserCloudCornerLastHandler()

/laser_cloud_corner_last トピックに紐付けられたコールバックである。それほど顕著ではないコーナー、あるいは顕著なコーナーとして分類された、 $k + 1$ 番目の点群を受け取って、そのタイムスタンプを `_timeLaserCloudCornerLast` にセットする。また ROS の点群メッセージ (`sensor_msgs::PointCloud2` 型) を、PCL(Point Cloud Library) における点群データ (`pcl::PointCloud` 型) に変換し、`BasicLaserMapping` の `_laserCloudCornerLast` メンバに格納する。

8.3 LaserMapping::laserCloudSurfLastHandler()

/laser_cloud_surf_last トピックに紐付けられたコールバックである。それほど顕著ではない平面、あるいは顕著な平面として分類された、 $k + 1$ 番目の点群を受け取って、そのタイムスタンプを `_timeLaserCloudSurfLast` にセットする。また ROS の点群メッセージ (`sensor_msgs::PointCloud2` 型) を、PCL(Point Cloud Library) における点群データ (`pcl::PointCloud` 型) に変換し、`BasicLaserMapping` の `_laserCloudSurfLast` メンバに格納する。

8.4 LaserMapping::laserCloudFullResHandler()

/velodyne_cloud_3 トピックに紐付けられたコールバックである。 $k + 1$ 番目の点群を受け取って、そのタイムスタンプを `_timeLaserCloudFullRes` にセットする。また ROS の点群メッセージ (`sensor_msgs::PointCloud2` 型) を、PCL(Point Cloud Library) における点群データ (`pcl::PointCloud` 型) に変換し、`BasicLaserMapping` の `_laserCloudFullRes` メンバに格納する。

8.5 LaserMapping::laserOdometryHandler()

/laser_odom_to_init トピックに紐付けられたコールバックである。オドメトリによって得られた、時刻 t_{k+2} における LiDAR の姿勢が laserOdometry->pose に格納されている。この姿勢は、BasicLaserOdometry::process() のなかで $T_{k+1}^S = [t_x^{k+1}, t_y^{k+1}, t_z^{k+1}, \theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}]^\top$ として登場したものである。

回転成分 laserOdometry->pose.orientation はクォータニオンで表現されているので、これをオイラー角 $\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}$ に戻す必要がある。オイラー角とクォータニオンの間で変換する際は、ROS と LOAM が採用している座標軸の違いに気をつけなければならない、それは LaserOdometry::publishResult() で説明した通りである。クォータニオン geoQuat を $(q_1, q_2, q_3, q_0) = (-q'_2, -q'_3, q'_1, q'_0)$ で表すと、これは LOAM の座標軸において、(1)Z 軸まわりに θ_z^{k+1} 、(2)X 軸まわりに θ_x^{k+1} 、(3)Y 軸まわりに θ_y^{k+1} という回転を表している。tf::Quaternion() によって $(q_3, -q_1, -q_2, q_0) = (q'_1, q'_2, q'_3, q'_0)$ を作成しているが、これは ROS の座標軸において、(1) \tilde{X} 軸まわりに θ_z^{k+1} 、(2) \tilde{Y} 軸まわりに $-\theta_x^{k+1}$ 、(3) \tilde{Z} 軸まわりに $-\theta_y^{k+1}$ という回転を表している。getRPY() によって得られるオイラー角は、ROS の座標軸まわりの回転角を表しているから、roll には θ_z^{k+1} 、pitch には $-\theta_x^{k+1}$ 、yaw には $-\theta_y^{k+1}$ が格納される。従って-pitch, -yaw, roll は $\theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}$ 、即ち LOAM の座標軸周りの回転角を表す。

BasicLaserMapping の updateOdometry() を呼び出して、受け取った姿勢 T_{k+1}^S を BasicLaserMapping の _transformSum メンバに格納する。また姿勢のタイムスタンプを _timeLaserOdometry に格納する。

8.6 LaserMapping::imuHandler()

/imu/data トピックに紐付けられたコールバックである。IMU データの回転角 (グローバル座標系からみた IMU 座標系の回転) はクォータニオンで表されるので、オイラー角 roll, pitch, yaw に変換する。roll, pitch, yaw は、ROS の座標軸まわりの回転角、即ち (1) \tilde{X} 軸まわりに roll、(2) \tilde{Y} 軸まわりに pitch、(3) \tilde{Z} 軸まわりに yaw という回転を表す。BasicLaserMapping の updateIMU() を呼び出して、タイムスタンプと、 \tilde{X}, \tilde{Y} 軸まわりの回転角を、リングバッファに追加する (\tilde{Z} 軸まわりの回転角は使われていない)。IMU データは、マッピングによって得た、時刻 t_{k+2} における LiDAR の姿勢 _transformAftMapped を補正するために利用される。

8.7 LaserMapping::hasNewData()

トピックを通じて新しいデータが届いているかどうかを確認する。Subscribe している 5 つのトピックのうち、IMU データを除く 4 つ全てのトピックから新しいデータが届いており、かつそれらのタイムスタンプが揃っている (新しいデータが、同一の点群に対応している) 場合に true を返す。

8.8 LaserMapping::publishResult()

LiDAR の周囲を表す点群 (BasicLaserMapping の _laserCloudSurroundDS) が更新された場合は、/laser_cloud_surround トピックから送出する。続いて、現在の ($k+1$ 番目の) 点群 (BasicLaserMapping の _laserCloudFullRes) を /velodyne_cloud_registered トピックから送出する。これらの点群に含まれる各点は、マッピングによって得られた、時刻 t_{k+2} における LiDAR の姿勢 (BasicLaserMapping の _

transformAftMapped) を使って、グローバル座標系に変換されている。

最後に、時刻 t_{k+2} におけるマッピング後の LiDAR の姿勢 `_transformAftMapped` を `/aft_mapped_to_init` トピックから送出する。これは `_transformAftMapped` に格納されているが、回転成分はオイラー角で表現され、LOAM の座標軸で定義される。一方 `/aft_mapped_to_init` トピックが Publish する `nav_msgs::Odometry` 型のメッセージは、回転成分をクォータニオンで表現し、こちらは ROS の座標系で定義される。LOAM の座標軸 X, Y, Z において定義されるオイラー角を、ROS の座標軸 $\tilde{X}, \tilde{Y}, \tilde{Z}$ において定義されるクォータニオンに変換しなければならないが、そのための方法は `BasicLaserOdometry` の `publishResult()` で説明した通りである。`_transformAftMapped(T_{k+1}^W)` がもつオイラー角を $\theta_x^{k+1,W}, \theta_y^{k+1,W}, \theta_z^{k+1,W}$ で表現する。`tf::createQuaternionMsgFromRollPitchYaw()` に $\theta_z^{k+1,W}, -\theta_x^{k+1,W}, -\theta_y^{k+1,W}$ をこの順で与えることで、ROS の座標軸について、(1) \tilde{X} 軸まわりに $\theta_z^{k+1,W}$ 、(2) \tilde{Y} 軸まわりに $-\theta_x^{k+1,W}$ 、(3) \tilde{Z} 軸まわりに $-\theta_y^{k+1,W}$ だけ回転させるようなクォータニオン (q'_1, q'_2, q'_3, q'_0) を作成する (変数 `geoQuat`)。この要素を入れ替えたクォータニオン ($-q'_2, -q'_3, q'_1, q'_0$) は、LOAM の座標系について、(1) Z 軸まわりに $\theta_z^{k+1,W}$ 、(2) X 軸まわりに $\theta_x^{k+1,W}$ 、(3) Y 軸まわりに $\theta_y^{k+1,W}$ だけ回転させるが、これが本来欲しかったものである (ROS では $\tilde{X}, \tilde{Y}, \tilde{Z}$ 、LOAM では Z, Y, X の順に回転させる)。

`nav_msgs::Odometry` 型のメッセージ `odomAftMapped` に以下の内容をセットして、`/laser_odom_to_init` トピックから Publish する。上で求めたクォータニオン ($-q'_2, -q'_3, q'_1, q'_0$) と、`_transformAftMapped(T_{k+1}^W)` の並進成分 $t_x^{k+1,W}, t_y^{k+1,W}, t_z^{k+1,W}$ を、pose フィールドにセットする。また、オドメトリによって得られた、時刻 t_{k+2} における LiDAR 座標系の姿勢 `_transformBefMapped(T_{k+1}^S)` を、twist フィールドにセットする (本来このフィールドは速度を意味するので、正しい使い方ではない)。twist フィールド (`geometry_msgs::Twist` 型) は、角速度 (angular フィールド) をオイラー角で表すので、先ほどのようにクォータニオンに変換する必要はない。続いて、`_transformAftMapped` が表している、フレーム間の座標関係を `/tf` トピックに Broadcast する。`_transformAftMapped` は、グローバル座標系 (camera_init フレーム) からみた、時刻 t_{k+2} におけるマッピング後の LiDAR 座標系 (`/aft_mapped` フレーム) の相対的な姿勢である。従って、親フレーム ID として `camera_init`、子フレーム ID として `/aft_mapped` を指定する。相対姿勢の並進成分を `setOrigin()`、回転成分を `setRotation()` により指定する。回転成分はクォータニオンによって表現されるので、先ほどのクォータニオン ($-q'_2, -q'_3, q'_1, q'_0$) を使えばよい。`/aft_mapped_to_init` トピックの姿勢データは、後続の `TransformMaintenance` ノードで使用される。

9 BasicTransformMaintenance クラス

9.1 BasicTransformMaintenance::transformAssociateMap()

このメソッドは `TransformMaintenance` の `laserOdometryHandler()` から呼び出されるので、その状況を前提に説明する。図 14 に示すように、各メンバ `_transformSum`、`_transformBefMapped`、`_transformAftMapped` には次の内容がセットされている。

- `_transformSum`: 時刻 t_{k+2} におけるオドメトリの姿勢 $T_{k+1}^S = [t_{k+1}^S, R_{k+1}^S]$ 。言い換えると、グローバル座標系 G からみた、時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 O_{k+2} の姿勢。
- `_transformBefMapped`: 時刻 t_{j+1} におけるオドメトリの姿勢 $T_j^S = [t_j^S, R_j^S]$ 。言い換えると、グローバル座標系 G からみた、時刻 t_{j+1} におけるオドメトリ後の LiDAR 座標系 O_{j+1} の姿勢。
- `_transformAftMapped`: 時刻 t_{j+1} におけるマッピング後の姿勢 $T_j^W = [t_j^W, R_j^W]$ 。言い換えると、グ

ローバル座標系 G からみた、時刻 t_{j+1} におけるマッピング後の LiDAR 座標系 M_{j+1} の姿勢。

この 3 つを使ってオドメトリとマッピングの結果を結合し、時刻 t_{k+2} における LiDAR の姿勢 $T_{k+1}^C = [t_{k+1}^C, R_{k+1}^C]$ を求める (グローバル座標系 G からみた、時刻 t_{k+2} における結合後の LiDAR 座標系 C_{k+2} の姿勢)。 T_{k+1}^C は `_transformMapped` に格納される。姿勢の並進成分 t_{k+1}^C と回転成分 R_{k+1}^C は次のように求められる。

$$R_{k+1}^C = R_j^W (R_j^S)^\top R_{k+1}^S \quad (178)$$

$$t_{k+1}^C = t_j^W - R_{k+1}^C (R_{k+1}^S)^\top (t_j^S - t_{k+1}^S) \quad (179)$$

$(R_j^S)^\top R_{k+1}^S$ は、点の座標を O_{k+2} から G , G から O_{j+1} へと変換する (座標軸を G から O_{k+2} , O_{j+1} から G へと変換する) 回転行列を合成したものであるから、点の座標を O_{k+2} から O_{j+1} , 座標軸を O_{j+1} から O_{k+2} へと移す。この回転 $(R_j^S)^\top R_{k+1}^S$ を、座標軸 M_{j+1} から C_{k+2} への回転とみなして、 G から M_{j+1} への座標軸の回転を表す行列 R_j^W と合成する。最終的に得られる $R_{k+1}^C = R_j^W (R_j^S)^\top R_{k+1}^S$ は、点の座標を C_{k+2} から M_{j+1} , M_{j+1} から G へと変換する (座標軸を M_{j+1} から C_{k+2} , G から M_{j+1} へと変換する) 回転行列の合成であるから、点の座標を C_{k+2} から G , 座標軸を G から C_{k+2} へと移す。

オドメトリによる、時刻 t_{k+2} から t_{j+1} までの LiDAR の位置変化は $t_j^S - t_{k+1}^S$ とグローバル座標系で書ける (t_j^S, t_{k+1}^S はいずれもグローバル座標系)。これに回転行列 $(R_{k+1}^S)^\top$ を適用することで、グローバル座標系から O_{k+2} 座標系へと変換できる。 $(R_{k+1}^S)^\top (t_j^S - t_{k+1}^S)$ は、時刻 t_{k+2} から t_{j+1} までの LiDAR の位置変化 (O_{k+2} 座標系から O_{j+1} 座標系への位置変化) を O_{k+2} 座標系から眺めたものであるが、 C_{k+2} 座標系から眺めたもの (C_{k+2} 座標系から M_{j+1} 座標系への位置変化) として解釈することで、オドメトリとマッピングの結果を結合できる。 C_{k+2} 座標系で表される、オドメトリによる位置変化 $(R_{k+1}^S)^\top (t_j^S - t_{k+1}^S)$ を、回転行列 R_{k+1}^C を使ってグローバル座標系に戻す。 $R_{k+1}^C (R_{k+1}^S)^\top (t_j^S - t_{k+1}^S)$ は、 C_{k+2} 座標系から M_{j+1} 座標系への位置変化を、グローバル座標系で表している。 G からみた M_{j+1} 座標系の位置 t_j^W に、 $-R_{k+1}^C (R_{k+1}^S)^\top (t_j^S - t_{k+1}^S)$, 即ち M_{j+1} 座標系から C_{k+2} 座標系への位置変化を足すことで、 G からみた C_{k+2} 座標系の位置 t_{k+1}^C が得られる。

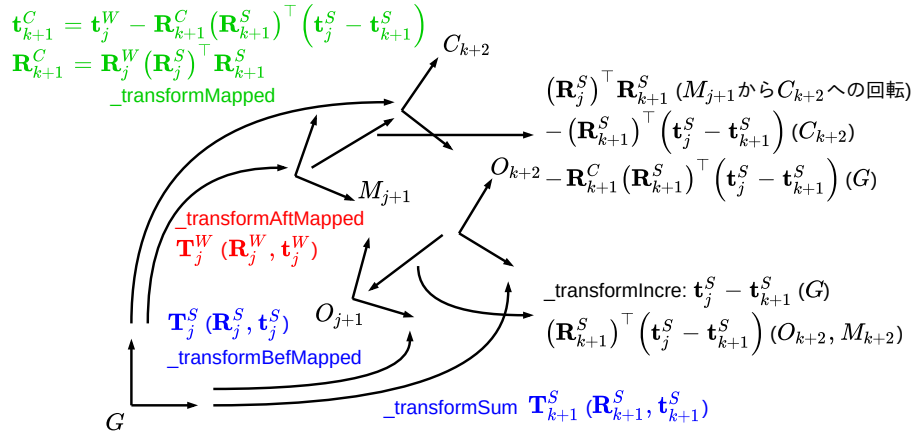
$R_{k+1}^C = R_y(\theta_y^{k+1,C})R_x(\theta_x^{k+1,C})R_z(\theta_z^{k+1,C})$ からオイラー角 $\theta_x^{k+1,C}, \theta_y^{k+1,C}, \theta_z^{k+1,C}$ を取り出す計算や、並進成分 $t_x^{k+1,C}, t_y^{k+1,C}, t_z^{k+1,C}$ の計算は、BasicLaserMapping の `transformAssociateMap()` と全く同様に行える。最初に、 $R_{k+1}^S = R_y(\theta_y^{k+1})R_x(\theta_x^{k+1})R_z(\theta_z^{k+1})$, $(R_j^S)^\top = R_z(-\theta_z^j)R_x(-\theta_x^j)R_y(-\theta_y^j)$, $R_j^W = R_y(\theta_y^{j,W})R_x(\theta_x^{j,W})R_z(\theta_z^{j,W})$ の各要素を書き出す。続いて、 $R_{k+1}^C = R_j^W (R_j^S)^\top R_{k+1}^S$ のうち 5 つの要素を計算し、逆三角関数を用いればオイラー角 $\theta_x^{k+1,C}, \theta_y^{k+1,C}, \theta_z^{k+1,C}$ を復元できる。

10 TransformMaintenance クラス

10.1 TransformMaintenance::setup()

TransformMaintenance ノードが Subscribe あるいは Publish するトピックを設定する。Subscribe するのは次の 2 つのトピックである。

- `/laser_odom_to_init` トピック (nav_msgs::Odometry 型): オドメトリによって得られた、時刻 t_{k+2} における LiDAR の姿勢 T_{k+1}^S (グローバル座標系)。言い換えると、グローバル座標系からみた、時刻 t_{k+2} におけるオドメトリ後の LiDAR 座標系 (図 14 の O_{k+2}) の姿勢。グローバル座標系は camera_



G : グローバル座標系
 O_{j+1} : オドメトリによって得た, 時刻 t_{j+1} における LiDAR 座標系
 O_{k+2} : オドメトリによって得た, 時刻 t_{k+2} における LiDAR 座標系
 M_{j+1} : マッピングによって得た, 時刻 t_{j+1} における LiDAR 座標系
 C_{k+2} : マッピングとオドメトリの姿勢の結合によって得た, 時刻 t_{k+2} における LiDAR 座標系

$$\mathbf{T}_k^S = [t_x^k, t_y^k, t_z^k, \theta_x^k, \theta_y^k, \theta_z^k]^\top$$

$$\mathbf{R}_k^S = \mathbf{R}_y(\theta_y^k) \mathbf{R}_x(\theta_x^k) \mathbf{R}_z(\theta_z^k)$$

$$\mathbf{t}_k^S = [t_x^k, t_y^k, t_z^k]^\top$$

$$\mathbf{T}_k^W = [t_x^{k,W}, t_y^{k,W}, t_z^{k,W}, \theta_x^{k,W}, \theta_y^{k,W}, \theta_z^{k,W}]^\top$$

$$\mathbf{R}_k^W = \mathbf{R}_y(\theta_y^{k,W}) \mathbf{R}_x(\theta_x^{k,W}) \mathbf{R}_z(\theta_z^{k,W})$$

$$\mathbf{t}_k^W = [t_x^{k,W}, t_y^{k,W}, t_z^{k,W}]^\top$$

$$\mathbf{T}_k^C = [t_x^{k,C}, t_y^{k,C}, t_z^{k,C}, \theta_x^{k,C}, \theta_y^{k,C}, \theta_z^{k,C}]^\top$$

$$\mathbf{R}_k^C = \mathbf{R}_y(\theta_y^{k,C}) \mathbf{R}_x(\theta_x^{k,C}) \mathbf{R}_z(\theta_z^{k,C})$$

$$\mathbf{t}_k^C = [t_x^{k,C}, t_y^{k,C}, t_z^{k,C}]^\top$$

図 14 transformAssociateMap() における座標系

init フレーム, オドメトリ後の LiDAR 座標系は/laser_odom フレームに相当する. LaserOdometry ノードから 10Hz の頻度で Publish される.

- /aft_mapped_to_init トピック (nav_msgs::Odometry 型): マッピングとオドメトリによって得られた, 時刻 t_{j+1} における LiDAR の姿勢 $\mathbf{T}_j^W, \mathbf{T}_j^S$ (グローバル座標系). 言い換えると, グローバル座標系から見た, 時刻 t_{j+1} におけるマッピング後, オドメトリ後の LiDAR 座標系 (図 14 の M_{j+1}, O_{j+1}) の姿勢. グローバル座標系は camera_init フレーム, マッピング後の LiDAR 座標系は/aft_mapped フレーム, オドメトリ後の LiDAR 座標系は/laser_odom フレームに相当する. LaserMapping ノードから 5Hz の頻度で Publish される.

/laser_odom_to_init トピックと/aft_mapped_to_init トピックは同期しておらず, 到着頻度も異なる. 2 つのトピックから届く姿勢のデータが, 別々の時刻のものであることを示すため, 時刻を j と k とで分けて表記している. また Publish するのは次のトピックである.

- /integrated_to_init トピック (nav_msgs::Odometry 型): /laser_odom_to_init トピックと/aft_mapped_to_init トピックから届いた姿勢を結合して得られた, 現在の時刻 t_{k+2} における姿勢 T_{k+2}^C (図 14 の C_{k+2} 座標系). 言い換えると, グローバル座標系からみた, 時刻 t_{k+2} における結合後の LiDAR 座標系 (図 14 の C_{k+2}) の姿勢. グローバル座標系は camera_init フレーム, 結合後の LiDAR 座標系は/camera フレームに相当する. /laser_odom_to_init トピックからデータが到着する度に, 言い換えると 10Hz の頻度で Publish される.

/laser_odom_to_init トピックからデータが届くと/integrated_to_init トピックから姿勢が Publish される. 姿勢が Publish される時点では, /laser_odom_to_init トピックから届いたデータが最新であり, /aft_mapped_to_init トピックから届いたデータはそれよりも古いはずである. オドメトリの結果を使ってマッピングを行うので, 時刻 t_{k+2} におけるオドメトリの姿勢 T_{k+1}^S が/laser_odom_to_init トピックから到着した時点では, マッピングによる姿勢 T_{k+1}^W はまだ計算途中である. /aft_mapped_to_init トピックからは, どんなに新しくとも, 時刻 t_{k+1} までのマッピングの姿勢 T_k^W とオドメトリの姿勢 T_k^S しか届いていないはずである. 従って, 上記の時刻の表記において, $j < k + 1$ が成り立つといえる.

10.2 TransformMaintenance::laserOdometryHandler()

/laser_odom_to_init トピックに紐付けられたコールバックである. updateOdometry() を呼び出して, 時刻 t_{k+2} におけるオドメトリの姿勢 $T_{k+1}^S = [t_x^{k+1}, t_y^{k+1}, t_z^{k+1}, \theta_x^{k+1}, \theta_y^{k+1}, \theta_z^{k+1}]^\top$ を _transformSum メンバに代入する.

transformAssociateMap() によって, マッピングによって得られた時刻 $j + 1$ における LiDAR の姿勢 T_j^W (_transformAftMapped) に, オドメトリによって得られた時刻 $j + 1$ から $k + 2$ までの LiDAR の姿勢変化 (T_j^S から T_{k+1}^S , _transformBefMapped から _transformSum) を結合することで, 時刻 $k + 2$ における LiDAR の姿勢 T_{k+1}^C (_transformMapped) を得る (図 14). _transformAftMapped, _transformBefMapped, _transformSum は, camera_init フレーム (図 14 の G) からみた, 時刻 t_{j+1} における/aft_mapped フレーム (M_{j+1}), 時刻 t_{j+1} における/laser_odom フレーム (O_{j+1}), 時刻 t_{k+2} における/laser_odom フレーム (O_{k+2}) の姿勢である. また transformMapped は, camera_init フレーム (G) からみた, 時刻 t_{k+2} における/camera フレーム (C_{k+2}) の姿勢である.

マッピングとオドメトリの結果を結合して得られた LiDAR の姿勢 T_{k+1}^C (/transformMapped) を, nav_msgs::Odometry 型のメッセージ_laserOdometry2 にセットして, /integrated_to_init トピックから Publish する. /transformMapped が表すフレーム間の座標変換を/tf トピックに Broadcast する. /transformMapped は camera_init フレームからみた/camera フレームの姿勢であるから, 親フレーム (frame_id_) に camera_init, 子フレーム (child_frame_id_) に/camera を指定する. クォータニオンとオイラー角との変換が必要であるが, その説明は LaserOdometry や LaserMapping の publishResult() で詳しく触れたので省略する.

10.3 TransformMaintenance::odomAftMappedHandler()

/aft_mapped_to_init トピックに紐付けられたコールバックである. updateMappingTransform() を呼び出して, 時刻 t_{j+1} におけるマッピング後の LiDAR の姿勢 T_j^W と, オドメトリ後の LiDAR の姿勢 T_j^S を, _transformAftMapped と _transformBefMapped メンバに代入する. ここでも, クォータニオンとオイラー

角との変換が登場する.

参考文献

- [1] J. Zhang and S. Singh, “LOAM: Lidar Odometry and Mapping in Real-time,” in *Proceedings of the Robotics: Science and Systems (RSS)*, July 2014.
- [2] J. Zhang, M. Kaess, and S. Singh, “On Degeneracy of Optimization-based State Estimation Problems,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 809–816, 2016.