Contents

Chapter 1

Notations

A large part of robotics is about developing machines that perceives and interact with the environment. For that robots use sensors to collect and process data, and knowing what the data describes is of utmost importance. Imagine obtaining the position of the robot but not knowing what that position is with respect to. Missing data descriptions such as what a position vector is expressing, what is it with respect to and more causes many hours of painful trail and error to extract that information.

In the following section the notation used throughout this document will be described, and it follows closely of that of Paul Furgale's [1]. The aim is to mitigate the ambiguity that arises when describing robot poses, sensor data and more.

A vector expressed in the world frame, $\mathcal{F}_{\mathrm{W}}$, is written as ${}_{\mathrm{W}}\mathbf{r}$. Or more precisely if the vector describes the position of the camera frame, $\mathcal{F}_{\mathrm{C}}$, expressed in $\mathcal{F}_{\mathrm{W}}$, the vector can be written as ${}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}$ with W and C as start and end points. The left hand subscripts indicates the coordinate system the vector is expressed in, while the right-hand subscripts indicate the start and end points. For brevity if the vector has the same start point as the frame to which it is expressed in, the same vector can be written as ${}_{\mathrm{W}}\mathbf{r}_{\mathrm{C}}$. Similarly a transformation of a point from $\mathcal{F}_{\mathrm{C}}$ to $\mathcal{F}_{\mathrm{W}}$ can be represented by a homogeneous transform matrix, $\mathbf{T}_{\mathrm{WC}}$, where its rotation matrix component is written as $\mathbf{C}_{\mathrm{WC}}$ and the translation component written as ${}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}$. A rotation matrix that is parametrized by quaternion $\mathbf{q}_{\mathrm{WC}}$ is written as $\mathbf{C}\{\mathbf{q}_{\mathrm{WC}}\}$.

Position:	${}_{\mathrm{W}}\mathbf{r}_{\mathrm{WB}}$
Velocity:	${}_{\mathrm{W}}\mathbf{v}_{\mathrm{WB}}$
Acceleration:	${}_{\mathrm{W}}\mathbf{a}_{\mathrm{WB}}$
Angular velocity:	${}_{\mathrm{W}}\boldsymbol{\omega}_{\mathrm{WB}}$
Rotation:	$\mathbf{C}_{\mathrm{WB}}$
Transform:	$\mathbf{T}_{\mathrm{WB}}$
Point:	${}_{\mathrm{W}}\mathbf{p}$

Chapter 2

Rotations

2.1 Euler Angles

Z-Y-X rotation sequence:

$$\mathbf{C}_{zyx} = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \tag{2.1}$$

2.2 Quaternions

A quaternion, $\mathbf{q} \in \mathbb{R}^4$, generally has the following form

$$\mathbf{q} = q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}, \tag{2.2}$$

where $\{q_w, q_x, q_y, q_z\} \in \mathbb{R}$ and $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ are the imaginary numbers satisfying

$$\begin{gathered} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \\ \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \;\; \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \;\; \mathbf{ki} = -\mathbf{ik} = \mathbf{j} \end{gathered} \tag{2.3}$$

corresponding to the Hamiltonian convention. The quaternion can be written as a 4 element vector consisting of a *real* (*scalar*) part, q_w, and *imaginary* (*vector*) part $\mathbf{q}_v$ as,

$$\mathbf{q} = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{2.4}$$

There are other quaternion conventions, for example, the JPL convention. A more detailed discussion between Hamiltonian and JPL quaternion convention is discussed in [2].

2.2.1 Main Quaternion Properties

Sum

Let $\mathbf{p}$ and $\mathbf{q}$ be two quaternions, the sum of both quaternions is,

$$\mathbf{p} \pm \mathbf{q} = \begin{bmatrix} p_w \\ \mathbf{p}_v \end{bmatrix} \pm \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_w \pm q_w \\ \mathbf{p}_v \pm \mathbf{q}_v \end{bmatrix}. \tag{2.5}$$

The sum between two quaternions $\mathbf{p}$ and $\mathbf{q}$ is **commutative** and **associative**.

$$\mathbf{p} + \mathbf{q} = \mathbf{q} + \mathbf{p} \tag{2.6}$$

$$\mathbf{p} + (\mathbf{q} + \mathbf{r}) = (\mathbf{p} + \mathbf{q}) + \mathbf{r} \tag{2.7}$$

Product

The quaternion multiplication (or product) of two quaternions $\mathbf{p}$ and $\mathbf{q}$, denoted by $\otimes$ is defined as

$$\mathbf{p} \otimes \mathbf{q} = (p_w + p_x\mathbf{i} + p_y\mathbf{j} + p_z\mathbf{k})(q_w + q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k}) \tag{2.8}$$

$$= \begin{matrix} (p_w q_w - p_x q_x - p_y q_y - p_z q_z) \\ (p_w q_x + p_x q_w + p_y q_z - p_z q_y) \quad \mathbf{i} \\ (p_w q_y - p_y q_w + p_z q_x + p_x q_z) \quad \mathbf{j} \\ (p_w q_z + p_z q_w - p_x q_y + p_y q_x) \quad \mathbf{k} \end{matrix} \tag{2.9}$$

$$= \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + q_x p_w + p_y q_z - p_z q_y \\ p_w q_y - p_y q_w + p_z q_x + p_x q_z \\ p_w q_z + p_z q_w - p_x q_y + p_y q_x \end{bmatrix} \tag{2.10}$$

$$= \begin{bmatrix} p_w q_w - \mathbf{p}_v^\top \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix}. \tag{2.11}$$

The quaternion product is **not commutative** in the general case[1],

$$\mathbf{p} \otimes \mathbf{q} \neq \mathbf{q} \otimes \mathbf{p} \ . \tag{2.12}$$

The quaternion product is however **associative**,

$$\mathbf{p} \otimes (\mathbf{q} \otimes \mathbf{r}) = (\mathbf{p} \otimes \mathbf{q}) \otimes \mathbf{r} \tag{2.13}$$

and **distributive over the sum**

$$\mathbf{p} \otimes (\mathbf{q} + \mathbf{r}) = \mathbf{p} \otimes \mathbf{q} + \mathbf{p} \otimes \mathbf{r} \quad \text{and} \quad (\mathbf{p} \otimes \mathbf{q}) + \mathbf{r} = \mathbf{p} \otimes \mathbf{r} + \mathbf{q} \otimes \mathbf{r} \tag{2.14}$$

The quaternion product can alternatively be expressed in matrix form as

$$\mathbf{p} \otimes \mathbf{q} = [\mathbf{p}]_L \mathbf{q} \quad \text{and} \quad \mathbf{p} \otimes \mathbf{q} = [\mathbf{q}]_R \mathbf{p} \ , \tag{2.15}$$

[1] There are exceptions to the general non-commutative rule, where either $\mathbf{p}$ or $\mathbf{q}$ is real such that $\mathbf{p}_v \times \mathbf{q}_v = 0$, or when both $\mathbf{p}_v$ and $\mathbf{q}_v$ are parallel, $\mathbf{p}_v || \mathbf{q}_v$. Only in these cirmcumstances is the quaternion product commutative.

where $[\mathbf{p}]_L$ and $[\mathbf{q}]_R$ are the left and right quaternion-product matrices which are derived from (2.10),

$$[\mathbf{p}]_L = \begin{bmatrix} p_w & -p_x & -p_y & -p_z \\ p_x & p_w & -p_z & p_y \\ p_y & p_z & p_w & -p_x \\ p_z & -p_y & p_x & p_w \end{bmatrix}, \quad \text{and} \quad [\mathbf{q}]_R = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix}, \tag{2.16}$$

or inspecting (2.11) a compact form can be derived as,

$$[\mathbf{p}]_L = \begin{bmatrix} 0 & -\mathbf{p}_v^\top \\ \mathbf{p}_w \mathbf{I}_{3\times 3} + \mathbf{p}_v & \mathbf{p}_w \mathbf{I}_{3\times 3} - \lfloor \mathbf{p}_v \ \times \rfloor \end{bmatrix} \tag{2.17}$$

and

$$[\mathbf{q}]_R = \begin{bmatrix} 0 & -\mathbf{q}_v^\top \\ \mathbf{q}_w \mathbf{I}_{3\times 3} + \mathbf{q}_v & \mathbf{q}_w \mathbf{I}_{3\times 3} - \lfloor \mathbf{q}_v \ \times \rfloor \end{bmatrix}, \tag{2.18}$$

where $\lfloor \bullet \ \times \rfloor$ is the skew operator that produces a matrix cross product matrix, and is defined as

$$\lfloor \mathbf{v} \ \times \rfloor = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \quad \mathbf{v} \in \mathbb{R}^3 \tag{2.19}$$

Conjugate

The conjugate operator for quaternion, $(\bullet)^*$, is defined as

$$\mathbf{q}^* = \begin{bmatrix} q_w \\ -\mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_w \\ -q_x \\ -q_y \\ -q_z \end{bmatrix}. \tag{2.20}$$

This has the properties

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = q_w^2 + q_x^2 + q_y^2 + q_z^2 = \begin{bmatrix} q_w^2 + q_x^2 + q_y^2 + q_z^2 \\ \mathbf{0} \end{bmatrix}, \tag{2.21}$$

and

$$(\mathbf{p} \otimes \mathbf{q})^* = \mathbf{q}^* \otimes \mathbf{p}^*. \tag{2.22}$$

Norm

The norm of a quaternion is defined by

$$\|\mathbf{q}\| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*} \tag{2.23}$$

$$= \sqrt{\mathbf{q}^* \otimes \mathbf{q}} \tag{2.24}$$

$$= \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \ \in \mathbb{R}, \tag{2.25}$$

and has the property

$$\|\mathbf{p} \otimes \mathbf{q}\| = \|\mathbf{q} \otimes \mathbf{p}\| = \|\mathbf{p}\| \|\mathbf{q}\| \tag{2.26}$$

2.2.2 Quaternion from Two Vectors

Using the properties of the cross and dot product

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\|\|\mathbf{v}\| \cos\theta \tag{2.27}$$

$$\|\mathbf{u} \times \mathbf{v}\| = \|\mathbf{u}\|\|\mathbf{v}\|\|\sin\theta\|, \tag{2.28}$$

the axis angle, $\boldsymbol{\theta} \in \mathbb{R}^3$, can be obtained from $\mathbf{u}$ and $\mathbf{v}$ with

$$\boldsymbol{\theta} = \theta \mathbf{e} \tag{2.29}$$

$$\theta = \cos^{-1}\left(\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}\right) \quad , \ \theta \in \mathbb{R} \tag{2.30}$$

$$\mathbf{e} = \frac{\mathbf{u} \times \mathbf{v}}{\|\mathbf{u} \times \mathbf{v}\|} \quad , \ \mathbf{e} \in \mathbb{R}^3 \tag{2.31}$$

where $\mathbf{e}$ is the unit vector that defines the rotation axis and θ is the rotation angle about $\mathbf{e}$. Once the axis angle, $\boldsymbol{\theta}$, is obtained a quaternion can be formed

$$\mathbf{q} = \cos\frac{\theta}{2} + \mathbf{i}\sin\frac{\theta}{2}e_x + \mathbf{j}\sin\frac{\theta}{2}e_y + \mathbf{k}\sin\frac{\theta}{2}e_z \ . \tag{2.32}$$

Example: Attitude from gravity and accelerometer vectors

In robotics knowing the attitude of the system is often required. An Inertial Measurement Unit (IMU) is commonly used to obtain this information. Using the method described previously, a gravity vector along with an accelerometer measurement vector can be used to obtain an attitude in form of a quaternion.

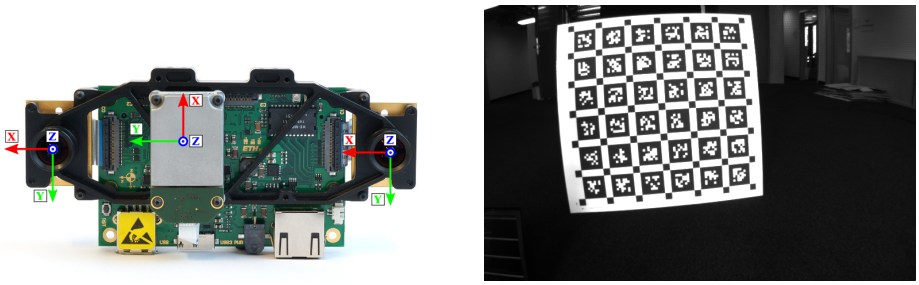
Let $\mathbf{g} \in \mathbb{R}^3$ be the gravity vector, and $\mathbf{a}_m \in \mathbb{R}^3$ be the accelerometer measurement from an IMU. With the two vectors $\mathbf{g}$ and $\mathbf{a}_m$ a quaternion $\mathbf{q}_{\text{WS}}$ expressing the rotation of the IMU sensor frame, $\mathcal{F}_{\text{S}}$, with respect to the world frame, $\mathcal{F}_{\text{W}}$, can be calculated given that values for $\mathbf{g}$ and $\mathbf{a}_m$ are known. For example let

$$\mathbf{g} = \begin{bmatrix} 0 & 0 & -9.81 \end{bmatrix}^\top \tag{2.33}$$

$$\mathbf{a}_m = \begin{bmatrix} 9.2681 & -0.310816 & -3.14984 \end{bmatrix}^\top, \tag{2.34}$$

taken from the first measurement of the `imu_april` calibration sequence of the EuRoC MAV dataset.

Before calculating the axis-angle, however, it should be noted that when an accelerometer is at rest the measurement reading in the z-axis is positive instead of negative. The reason is accelerometers measures acceleration by measuring the displacement of a proof mass that is suspended with springs. For example, if gravity is ignored and the accelerometer moves upwards, the proof mass will be displaced towards the bottom of the accelerometer. This is interpreted as an acceleration in the upwards direction, and so when the accelerometer is at rest on a flat surface, gravity pulls on the proof mass yeilding a positive measurement



(a) VI-Sensor Coordinate Frames (b) Image frame

Figure 2.1: EuRoC MAV Dataset - `imu_april` Sequence

in the upwards direction. To resolve this issue the gravity vector is negated, and so $\mathbf{u} = -\mathbf{g}$ and $\mathbf{v} = \mathbf{a}_m$. Using (2.30) and (2.31) the axis-angle, $\boldsymbol{\theta}$, is thereby

$$\theta = 1.8982 \tag{2.35}$$

$$\mathbf{e} = \begin{bmatrix} 0.03352 & 0.99944 & 0.00000 \end{bmatrix}^\top \tag{2.36}$$

Finally the quaternion, $\mathbf{q}_{\text{WS}}$, can be calculated using (2.32) resulting in

$$\mathbf{q}_{\text{WS}} = \begin{bmatrix} 0.58240 & 0.02725 & 0.81245 & 0.00000 \end{bmatrix}^\top . \tag{2.37}$$

2.2.3 Quaternion to Rotation Matrix

$$\mathbf{R}\{\mathbf{q}\} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_y - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{2.38}$$

2.3 Differential Calculus

Lie Group $SO(3)$

- Not a vector space
- Has no addition operator
- Has no subtraction operator

State estimation frameworks rely on small differences and gradients in order to correct the state estimate. Orientations unlike translation and velocity do not have an addition operator, as such it is more involving to update or find the gradients of orientations. Forunately, since orientations are a special orhogonal group $SO(3)$ as well as a Lie group, an exponential map exists that relates to its Lie algebra allowing orientations to be perturbed and its gradients calculated.

Elements in Lie algebra are abstract vectors and not suitable for actual computations. A basis $\mathbf{B} = [\vec{\boldsymbol{\varphi}}_1 \;\; \vec{\boldsymbol{\varphi}}_2 \;\; \vec{\boldsymbol{\varphi}}_3]$ can be used to extend the map to $\mathbb{R}^3$.

The definition of an exponential map $\exp : \mathbb{R}^3 \mapsto SO(3)$ of a coordinate tuple $\boldsymbol{\varphi}(\varphi_1, \varphi_2, \varphi_3) \in \mathbb{R}^3$ is defined by

$$\exp(\boldsymbol{\varphi}) := Exp(\vec{\boldsymbol{\varphi}}_1\varphi_1, \vec{\boldsymbol{\varphi}}_2\varphi_2, \vec{\boldsymbol{\varphi}}_3\varphi_3) \tag{2.39}$$

$\forall t, s \in \mathbb{R}$, $\forall \vec{\boldsymbol{\varphi}} \in$

$$Exp((t+s)\vec{\boldsymbol{\varphi}}) = Exp(t\vec{\boldsymbol{\varphi}}) \circ Exp(s\vec{\boldsymbol{\varphi}}) \tag{2.40}$$

$$\begin{aligned} \boxplus : & SO(3) \times \mathbb{R}^3 \to SO(3), & (2.41)\\ & \Phi, \boldsymbol{\varphi} \mapsto \exp(\boldsymbol{\varphi}) \circ \Phi, \\ \boxminus : & SO(3) \times SO(3) \to \mathbb{R}^3, & (2.42)\\ & \Phi_1, \Phi_2 \mapsto \log(\Phi_1 \circ \Phi_2^{-1}) \end{aligned}$$

Similar to regular addition and subtraction, both operators have the following identities,

$$\Phi \boxplus \mathbf{0} = \Phi \tag{2.43}$$

$$(\Phi \boxplus \boldsymbol{\varphi}) \boxminus \Phi = \boldsymbol{\varphi} \tag{2.44}$$

$$\Phi_1 \boxplus (\Phi_2 \boxminus \Phi_1) = \Phi_2 \tag{2.45}$$

Chapter 3

Computer Vision

3.0.1 Point on Line

$$\mathbf{x}^\top \mathbf{l} = 0 \tag{3.1}$$

3.0.2 Intersection of Lines

$$\mathbf{l}(\mathbf{l} \times \mathbf{l}')\mathbf{l}'(\mathbf{l} \times \mathbf{l}') = 0 \mathbf{l}^\top \mathbf{x} = \mathbf{l}^{\top\prime} \mathbf{x} = 0 \tag{3.2}$$

$$x = \mathbf{l} \times \mathbf{l}' \tag{3.3}$$

3.0.3 Plane

- A plane can be defined by the join between three points, or the join between a line and a point in general
- Two planes intersecting a unique line
- Three planes intersecting a unique point

Three Points Define a Plane

Suppose you have three points $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, and are incident with a plane, π then each point satisfies

$$\pi^\top \mathbf{X}_i = 0. \tag{3.4}$$

By stacking each point as a matrix

$$\begin{bmatrix} \mathbf{X}_1^\top \\ \mathbf{X}_2^\top \\ \mathbf{X}_3^\top \end{bmatrix} \pi = 0 \tag{3.5}$$

Since three points in general rare linearly independent, it follows that the $3x4$ matrix compsed of the points $\mathbf{X}_i$ as rows has rank 3.

3.1 Pinhole Camera Model

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

3.2 Radial Tangential Distortion

$$\begin{aligned} k_{\text{radial}} &= 1 + (k_1 r^2) + (k_2 r^4) \\ x' &= x \cdot k_{\text{radial}} \\ y' &= y \cdot k_{\text{radial}} \\ x'' &= x' + (2p_1 xy + p_2(r^2 + 2x^2)) \\ y'' &= y' + (p_1(r^2 + 2y^2) + 2p_2 xy) \end{aligned} \tag{3.7}$$

3.3 Equi-distant Distortion

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ \theta &= \arctan(r) \\ \theta_d &= \theta(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8) \\ x' &= (\theta_d / r) \cdot x \\ y' &= (\theta_d / r) \cdot y \end{aligned} \tag{3.8}$$

3.4 Feature Estimation

In the previous section we have discussed methods to detect and match between different camera frames. In this section we introduce a method to estimate feature positions in 3D using 2D pixels measurements of the same feature in different camera frames.

3.4.1 Feature Initialization

There are various methods for initializing the feature position. The linear triangulation method [3] is frequently used. This method assumes a pair of homogeneous pixel measurements $\mathbf{z}$ and $\mathbf{z}' \in \mathbb{R}^3$ that observes the same feature, $\mathbf{X}$, in homogeneous coordinates from two different camera frames. The homogeneous projection from 3D to 2D with a known camera matrix $\mathbf{P} \in \mathbb{R}^{3\times4}$ for each measurement is given as,

$$\begin{aligned} \mathbf{z} &= \mathbf{P}\mathbf{X} \\ \mathbf{z}' &= \mathbf{P}'\mathbf{X}. \end{aligned} \tag{3.9}$$

These equations can be combined to form a system of equations of the form $\mathbf{Ax} = 0$. To eliminate the homogeneous scale factor we apply a cross product to give three equations for each image point, for example $\mathbf{z} \times (\mathbf{PX}) = \mathbf{0}$ writing this out gives

$$\begin{aligned} x(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{1T}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{2T}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2T}\mathbf{X}) - y(\mathbf{p}^{1T}\mathbf{X}) &= 0 \end{aligned} \tag{3.10}$$

where $\mathbf{p}^{iT}$ is the i^{th} row of $\mathbf{P}$.

From Eq. (3.10), an equation of the form $\mathbf{Ax} = \mathbf{0}$ for each image point can be formed, where $\mathbf{x}$ represents the unknown homogeneous feature location to be estimated, and $\mathbf{A}$ is given as

$$\mathbf{A} = \begin{bmatrix} x(\mathbf{p}^{3T}) - (\mathbf{p}^{1T}) \\ y(\mathbf{p}^{3T}) - (\mathbf{p}^{2T}) \\ x'(\mathbf{p}'^{3T}) - (\mathbf{p}'^{1T}) \\ y'(\mathbf{p}'^{3T}) - (\mathbf{p}'^{2T}) \end{bmatrix} \tag{3.11}$$

giving a total of four equations in four homogeneous unknowns. Solving for $\mathbf{A}$ using SVD allows us to estimate the initial feature location.

3.4.2 Feature Refinement

The linear triangulation provides an initial feature position. In an ideal world, feature positions can be solved as a system of equations. In reality, however, errors are present in the camera poses and pixel measurements. The pixel measurements observing the same 3D point are generally noisy. In addition, the pinhole camera model and radial-tangential distortion model do not perfectly model the camera projection or distortion observed. Therefore an iterative method can be used to further refine the feature position. This problem is generally formulated as a non-linear least square problem and can be solved by numerical methods, such as the Gauss-Newton algorithm.

Let us consider the case where two different cameras at known locations observe the same feature in the scene. Our goal is to optimize for the feature position. With an i^{th} feature measurement, and corresponding i^{th} camera orientation as a quaternion, $\mathbf{q}_{\mathrm{C}_i\mathrm{W}}$ and i^{th} camera position, $\mathbf{r}_{\mathrm{C}_i}^{\mathrm{C}_i\mathrm{W}}$, we can formulate the re-projection error as,

$$\mathbf{e}_i(\boldsymbol{\theta}, \mathbf{q}_{\mathrm{C}_i\mathrm{W}}, \mathbf{r}_{\mathrm{C}_i}^{\mathrm{C}_i\mathrm{W}}) = \mathbf{z}_i - \mathbf{h}(\boldsymbol{\theta}, \mathbf{q}_{\mathrm{C}_i\mathrm{W}}, \mathbf{r}_{\mathrm{C}_i}^{\mathrm{C}_i\mathrm{W}}). \tag{3.12}$$

where the parameter, $\boldsymbol{\theta} \in \mathbb{R}^3$, represents the feature location to be optimized, $\mathbf{z}_i \in \mathbb{R}^2$ is the i-th pixel measurement, and $\mathbf{h} : \mathbb{R}^{10} \mapsto \mathbb{R}^2$ is the projection function that projects $\boldsymbol{\theta}$ into the measurement space using known camera extrinsics. The re-projection error is a geometric error corresponding to the Euclidean distance between measured and projected features onto the image plane, it is

used for quantifying the error of the estimated feature location. With the error function, $\mathbf{e}_i$, for one measurement, a cost function, $\mathbf{f}$, for m number of pixel measurements in the form of sum of squares can be defined as,

$$\mathbf{f}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \mathbf{e}_i(\boldsymbol{\theta}, \mathbf{q}_{\mathrm{C}_i\mathrm{W}}, \mathbf{r}_{\mathrm{C}_i}^{\mathrm{C}_i\mathrm{W}})^T \mathbf{e}_i(\boldsymbol{\theta}, \mathbf{q}_{\mathrm{C}_i\mathrm{W}}, \mathbf{r}_{\mathrm{C}_i}^{\mathrm{C}_i\mathrm{W}}). \tag{3.13}$$

Now that the cost function, $\mathbf{f}(\boldsymbol{\theta})$, is defined an unconstrained optimization can be performed using the cost function Eq. 3.13 to estimate the optimal feature position, $\boldsymbol{\theta}^*$, which minimizes the re-projection error over the set of collected measurements.

$$\boldsymbol{\theta}^* = \theta \mathbf{f}(\boldsymbol{\theta}). \tag{3.14}$$

3.5 Bundle Adjustment

Let $\mathbf{z} \in^2$ be the image measurement and $\mathbf{h}(\cdot) \in^2$ be the projection function that produces an image projection $\tilde{\mathbf{z}}$. The reprojection error e is defined as the euclidean distance between $\mathbf{z}$ and $\tilde{\mathbf{z}}$.

$$\mathbf{e} = \mathbf{z} - \tilde{\mathbf{z}} \tag{3.15}$$

Our aim given image measurement $\mathbf{z}$ is to find the image projection $\tilde{\mathbf{z}}$ that minimizes the reprojection e. The image projection $\tilde{\mathbf{z}}$ in pixels can be represented in homogeneous coordinates with u, v, w as

$$\tilde{\mathbf{z}} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} u/w \\ v/w \end{bmatrix} \tag{3.16}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P} \begin{bmatrix} {}_{\mathrm{W}}\mathbf{p} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{C}_{\mathrm{WC}}[{}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}] \tag{3.17}$$

where u, v, w is computed by projecting a landmark position ${}_{\mathrm{W}}\mathbf{p}$ in the world frame to the camera's image plane with projection matrix $\mathbf{P}$. The projection matrix, $\mathbf{P}$, can be decomposed into the camera intrinsics matrix, $\mathbf{K}$, the camera rotation, $\mathbf{C}_{\mathrm{WC}}$, and camera position, ${}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}$, expressed in the world frame. The orientation of the camera in (3.17) is represented using a rotation matrix. To reduce the optimization parameters the rotation matrix can be parameterized by a quaternion by using the following formula,

$$\mathbf{C}_{\mathrm{WC}} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}. \tag{3.18}$$

By parameterzing the rotation matrix with a quaternion, the optimization parameters for the camera's orientation is reduced from 9 to 4.

Our objective is to optimize for the camera rotation $\mathbf{C}_{\text{WC}}$, camera position ${}_{\text{W}}\mathbf{r}_{\text{WC}}$ and 3D landmark position ${}_{\text{W}}\mathbf{p}$ in order to minimize the cost function,

$$\underset{\mathbf{C}_{\text{WC}}, {}_{\text{W}}\mathbf{r}_{\text{WC}}, {}_{\text{W}}\mathbf{p}}{\operatorname{argmin}} \left\| \mathbf{z} - \mathbf{h}(\mathbf{C}_{\text{WC}}, {}_{\text{W}}\mathbf{r}_{\text{WC}}, {}_{\text{W}}\mathbf{p}) \right\|^2 \tag{3.19}$$

$$\underset{\mathbf{C}_{\text{WC}}, {}_{\text{W}}\mathbf{r}_{\text{WC}}, {}_{\text{W}}\mathbf{p}}{\operatorname{argmin}} \left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix} \right\|^2. \tag{3.20}$$

The cost function above assumes only a single measurement, if there are N measurements corresponding to N unique landmarks the cost function can be rewritten as a maximum likelihood estimation problem as,

$$\underset{\mathbf{C}_{\text{WC}}, {}_{\text{W}}\mathbf{r}_{\text{WC}}, {}_{\text{W}}\mathbf{p}}{\operatorname{argmin}} \sum_{j=1}^{N} \left\| \mathbf{z}_j - \mathbf{h}({\mathbf{C}_{\text{WC}}}^j, {}_{\text{W}}{\mathbf{r}_{\text{WC}}}^j, {}_{\text{W}}\mathbf{p}) \right\|^2 \tag{3.21}$$

under the assumption that the observed landmark, ${}_{\text{W}}\mathbf{p}$, measured in the image plane, z, are corrupted by a **zero-mean Gaussian noise**.

For the general case of M images taken at different camera poses the cost function can be further extended to,

$$\min_{\mathbf{C}_{\text{WC}}, {}_{\text{W}}\mathbf{r}_{\text{WC}}, {}_{\text{W}}\mathbf{p}} \sum_{i=1}^{M} \sum_{j=1}^{N} \left\| \mathbf{z}_{i,j} - \mathbf{h}({\mathbf{C}_{\text{WC}}}^i, {}_{\text{W}}{\mathbf{r}_{\text{WC}}}^i, {}_{\text{W}}\mathbf{p}) \right\|^2 \tag{3.22}$$

The optimization process begins by setting the first image camera pose as world origin, and subsequent $\mathbf{C}_{\text{WC}i}$ and ${}_{\text{W}}\mathbf{r}_{\text{WC}i}$ will be relative to the first camera pose.

Jacobians

The Jacobian for the optimization problem for a **single measurement** has the form:

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{h}}{\partial \mathbf{C}_{\text{WC}}} & \dfrac{\partial \mathbf{C}_{\text{WC}}}{\partial \mathbf{q}} & \dfrac{\partial \mathbf{h}}{\partial {}_{\text{W}}\mathbf{r}_{\text{WC}}} & \dfrac{\partial \mathbf{h}}{\partial {}_{\text{W}}\mathbf{p}} \end{bmatrix} \tag{3.23}$$

If there are two measurements the Jacobian is stacked with the following pattern:

$$\mathbf{J} = \begin{bmatrix} \text{Image } 1_{2\times 7} & \mathbf{0}_{2\times 7} & \text{3D Point}_{2\times 3} \\ \mathbf{0}_{2\times 7} & \text{Image } 2_{2\times 7} & \text{3D Point}_{2\times 3} \end{bmatrix} \tag{3.24}$$

Derivation for $\dfrac{\partial \mathbf{h}}{\partial \mathbf{C}_{\text{WC}}}$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{C}_{\text{WC}}} = \begin{bmatrix} \dfrac{w\dfrac{\partial u}{\partial \mathbf{C}_{\text{WC}}} - u\dfrac{\partial w}{\partial \mathbf{C}_{\text{WC}}}}{w^2} \\ \dfrac{w\dfrac{\partial v}{\partial \mathbf{C}_{\text{WC}}} - v\dfrac{\partial w}{\partial \mathbf{C}_{\text{WC}}}}{w^2} \end{bmatrix}_{2\times 9} \tag{3.25}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\mathbf{C}_{\mathrm{WC}}[{}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}]$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} [{}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}]$$

$$\frac{\partial u}{\partial \mathbf{C}_{\mathrm{WC}}} = \begin{bmatrix} f_x({}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}) & \mathbf{0}_{1\times 3} & p_x({}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}) \end{bmatrix}_{1\times 9} \tag{3.26}$$

$$\frac{\partial v}{\partial \mathbf{C}_{\mathrm{WC}}} = \begin{bmatrix} \mathbf{0}_{1\times 3} & f_y({}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}) & p_y({}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}) \end{bmatrix}_{1\times 9} \tag{3.27}$$

$$\frac{\partial w}{\partial \mathbf{C}_{\mathrm{WC}}} = \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & ({}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}) \end{bmatrix}_{1\times 9} \tag{3.28}$$

Derivation for $\frac{\partial \mathbf{C}_{\mathrm{WC}}}{\partial \mathbf{q}}$

$$\frac{\partial \mathbf{C}_{\mathrm{WC}}}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial \mathbf{C}_{\mathrm{WC}11}}{\partial \mathbf{q}} \\ \frac{\partial \mathbf{C}_{\mathrm{WC}12}}{\partial \mathbf{q}} \\ \vdots \\ \frac{\partial \mathbf{C}_{\mathrm{WC}33}}{\partial \mathbf{q}} \end{bmatrix}_{9\times 4} \tag{3.29}$$

$$\mathbf{C}_{\mathrm{WC}} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}$$

$$\frac{\mathbf{C}_{\mathrm{WC}11}}{\partial \mathbf{q}} = \begin{bmatrix} 0 & -4q_y & -4q_z & 0 \end{bmatrix} \tag{3.30}$$

$$\frac{\mathbf{C}_{\mathrm{WC}12}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_y & 2q_x & -2q_w & -2q_z \end{bmatrix} \tag{3.31}$$

$$\frac{\mathbf{C}_{\mathrm{WC}13}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_z & 2q_w & 2q_x & 2q_y \end{bmatrix} \tag{3.32}$$

$$\frac{\mathbf{C}_{\mathrm{WC}21}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_y & 2q_x & 2q_w & 2q_z \end{bmatrix} \tag{3.33}$$

$$\frac{\mathbf{C}_{\mathrm{WC}22}}{\partial \mathbf{q}} = \begin{bmatrix} 4q_x & 0 & 4q_z & 0 \end{bmatrix} \tag{3.34}$$

$$\frac{\mathbf{C}_{\mathrm{WC}23}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_w & 2q_z & 2q_y & 2q_x \end{bmatrix} \tag{3.35}$$

$$\frac{\mathbf{C}_{\mathrm{WC}31}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_z & -2q_w & 2q_x & -2q_y \end{bmatrix} \tag{3.36}$$

$$\frac{\mathbf{C}_{\mathrm{WC}32}}{\partial \mathbf{q}} = \begin{bmatrix} -4q_x & -4q_y & 0 & 0 \end{bmatrix} \tag{3.37}$$

$$\frac{\mathbf{C}_{\mathrm{WC}33}}{\partial \mathbf{q}} = \begin{bmatrix} 2q_w & 2q_z & 2q_y & 2q_x \end{bmatrix} \tag{3.38}$$

Derivation for $\dfrac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{p}}$

$$\frac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{p}} = \begin{bmatrix} \dfrac{w\dfrac{\partial u}{\partial {}_{\mathrm{W}}\mathbf{p}} - u\dfrac{\partial w}{\partial {}_{\mathrm{W}}\mathbf{p}}}{w^2} \\ \dfrac{w\dfrac{\partial v}{\partial {}_{\mathrm{W}}\mathbf{p}} - v\dfrac{\partial w}{\partial {}_{\mathrm{W}}\mathbf{p}}}{w^2} \end{bmatrix}_{2\times 3} \tag{3.39}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\mathbf{C}_{\mathrm{WC}}[{}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}]$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} [{}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}]$$

$$\frac{\partial u}{\partial {}_{\mathrm{W}}\mathbf{p}} = \begin{bmatrix} f_x\mathbf{C}_{\mathrm{WC}11} + c_x\mathbf{C}_{\mathrm{WC}31} & f_x\mathbf{C}_{\mathrm{WC}12} + c_x\mathbf{C}_{\mathrm{WC}32} & f_x\mathbf{C}_{\mathrm{WC}13} + c_x\mathbf{C}_{\mathrm{WC}33} \end{bmatrix} \tag{3.40}$$

$$\frac{\partial v}{\partial {}_{\mathrm{W}}\mathbf{p}} = \begin{bmatrix} f_y\mathbf{C}_{\mathrm{WC}21} + c_y\mathbf{C}_{\mathrm{WC}31} & f_y\mathbf{C}_{\mathrm{WC}22} + c_y\mathbf{C}_{\mathrm{WC}32} & f_y\mathbf{C}_{\mathrm{WC}23} + c_y\mathbf{C}_{\mathrm{WC}33} \end{bmatrix} \tag{3.41}$$

$$\frac{\partial w}{\partial {}_{\mathrm{W}}\mathbf{p}} = \begin{bmatrix} \mathbf{C}_{\mathrm{WC}31} & \mathbf{C}_{\mathrm{WC}32} & \mathbf{C}_{\mathrm{WC}33} \end{bmatrix} \tag{3.42}$$

Derivation for $\dfrac{\partial h}{\partial {}_{\mathbf{W}}\mathbf{r}_{\mathbf{WC}}}$

$$\frac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}} = \begin{bmatrix} \dfrac{w\dfrac{\partial u}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}} - u\dfrac{\partial w}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}}}{w^2} \\ \dfrac{w\dfrac{\partial v}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}} - v\dfrac{\partial w}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}}}{w^2} \end{bmatrix}_{2\times 3} \tag{3.43}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\mathbf{C}_{\mathrm{WC}}[{}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}]$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} [{}_{\mathrm{W}}\mathbf{p} - {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}]$$

$$\frac{\partial u}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}} = -\begin{bmatrix} f_x\mathbf{C}_{\mathrm{WC}11} + c_x\mathbf{C}_{\mathrm{WC}31} & f_x\mathbf{C}_{\mathrm{WC}12} + c_x\mathbf{C}_{\mathrm{WC}32} & f_x\mathbf{C}_{\mathrm{WC}13} + c_x\mathbf{C}_{\mathrm{WC}33} \end{bmatrix} \tag{3.44}$$

$$\frac{\partial v}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}} = -\begin{bmatrix} f_y\mathbf{C}_{\mathrm{WC}21} + c_y\mathbf{C}_{\mathrm{WC}31} & f_y\mathbf{C}_{\mathrm{WC}22} + c_y\mathbf{C}_{\mathrm{WC}32} & f_y\mathbf{C}_{\mathrm{WC}23} + c_y\mathbf{C}_{\mathrm{WC}33} \end{bmatrix} \tag{3.45}$$

$$\frac{\partial w}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WC}}} = -\begin{bmatrix} \mathbf{C}_{\mathrm{WC}31} & \mathbf{C}_{\mathrm{WC}32} & \mathbf{C}_{\mathrm{WC}33} \end{bmatrix} \tag{3.46}$$

Chapter 4

Calibration

$$\underset{\mathbf{T}_{\mathrm{WS}},\mathbf{T}_{\mathrm{SC}},\mathbf{T}_{\mathrm{WF}}}{\operatorname{argmin}} \left\|\mathbf{f}(\mathbf{T}_{\mathrm{WS}},\mathbf{T}_{\mathrm{SC}},\mathbf{T}_{\mathrm{WF}})\right\|^2 \tag{4.1}$$

Where

$$\mathbf{f}(\mathbf{T}_{\mathrm{WS}},\mathbf{T}_{\mathrm{SC}},\mathbf{T}_{\mathrm{WF}}) = \mathbf{z} - \mathbf{h}(\mathbf{T}_{\mathrm{WS}},\mathbf{T}_{\mathrm{SC}},\mathbf{T}_{\mathrm{WF}}) \tag{4.2}$$

In ceres-solver the expression $\left\|\mathbf{f}(\mathbf{T}_{\mathrm{WS}},\mathbf{T}_{\mathrm{SC}},\mathbf{T}_{\mathrm{WF}})\right\|^2$ is known as a residual block. As such the optimization jacobian matrix is with respect to the residual.

$$\hat{\mathbf{z}} = \mathbf{h}(\mathbf{T}_{\mathrm{WS}},\mathbf{T}_{\mathrm{SC}},\mathbf{T}_{\mathrm{WF}}) \tag{4.3}$$

$$\mathbf{e} = \mathbf{z} - \hat{\mathbf{z}} \tag{4.4}$$

$$\hat{\mathbf{z}} = \begin{bmatrix} {}_{\mathrm{C}}X/{}_{\mathrm{C}}Z \\ {}_{\mathrm{C}}Y/{}_{\mathrm{C}}Z \end{bmatrix} \tag{4.5}$$

$${}_{\mathrm{C}}\mathbf{p} = \begin{bmatrix} {}_{\mathrm{C}}X \\ {}_{\mathrm{C}}Y \\ {}_{\mathrm{C}}Z \end{bmatrix} \tag{4.6}$$

$$\frac{\partial \mathbf{h}}{\partial {}_{\mathrm{C}}\mathbf{p}} = \begin{bmatrix} 1/{}_{\mathrm{C}}Z & 0 & -{}_{\mathrm{C}}X/{}_{\mathrm{C}}Z^2 \\ 0 & 1/{}_{\mathrm{C}}Z & -{}_{\mathrm{C}}Y/{}_{\mathrm{C}}Z^2 \end{bmatrix} \tag{4.7}$$

$${}_{\mathrm{C}}\mathbf{p} = {\mathbf{T}_{\mathrm{SC}}}^{-1}\ {\mathbf{T}_{\mathrm{WS}}}^{-1}\ \mathbf{T}_{\mathrm{WF}}\ {}_{\mathrm{F}}\mathbf{p} \tag{4.8}$$

4.1 Jacobian w.r.t Sensor Pose, $\mathbf{T}_{\mathrm{WS}}$

$${}_{\mathrm{W}}\mathbf{p} = \mathbf{T}_{\mathrm{WS}}\ {}_{\mathrm{S}}\mathbf{p} \tag{4.9}$$

$$= \mathbf{C}_{\mathrm{WS}}\ {}_{\mathrm{S}}\mathbf{p} + {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WS}} \tag{4.10}$$

$$\frac{\partial \mathbf{h}}{\partial_{\mathrm{W}}\mathbf{p}} = \frac{\partial \mathbf{h}}{\partial_{\mathrm{C}}\mathbf{p}} \frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{W}}\mathbf{p}} \tag{4.11}$$

$$\frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{W}}\mathbf{p}} = \mathbf{C}_{\mathrm{CW}} \tag{4.12}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{T}_{\mathrm{WS}}} = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \delta\boldsymbol{\theta}} & \frac{\partial \mathbf{h}}{\partial_{\mathrm{W}}\mathbf{r}_{\mathrm{WS}}} \end{bmatrix} \tag{4.13}$$

$$\frac{\partial \mathbf{h}}{\partial \delta\boldsymbol{\theta}} = \frac{\partial \mathbf{h}}{\partial_{\mathrm{C}}\mathbf{p}} \frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{W}}\mathbf{p}} \frac{\partial_{\mathrm{W}}\mathbf{p}}{\partial \delta\boldsymbol{\theta}} \tag{4.14}$$

$$\frac{\partial_{\mathrm{W}}\mathbf{p}}{\partial \delta\boldsymbol{\theta}} = -\lfloor \mathbf{C}\{\delta\boldsymbol{\theta}\}\ {}_{\mathrm{S}}\mathbf{p}\ \times \rfloor \tag{4.15}$$

$$\frac{\partial \mathbf{h}}{\partial_{\mathrm{W}}\mathbf{r}_{\mathrm{WS}}} = \frac{\partial \mathbf{h}}{\partial_{\mathrm{C}}\mathbf{p}} \frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{W}}\mathbf{p}} \frac{\partial_{\mathrm{W}}\mathbf{p}}{\partial_{\mathrm{W}}\mathbf{r}_{\mathrm{WS}}} \tag{4.16}$$

$$\frac{\partial_{\mathrm{W}}\mathbf{p}}{\partial_{\mathrm{W}}\mathbf{r}_{\mathrm{WS}}} = \mathbf{I} \tag{4.17}$$

4.2 Jacobian w.r.t Sensor-Camera Extrinsics, $\mathbf{T}_{\mathbf{SC}}$

$${}_{\mathrm{S}}\mathbf{p} = \mathbf{T}_{\mathrm{SC}}\ {}_{\mathrm{C}}\mathbf{p} \tag{4.18}$$

$$= \mathbf{C}_{\mathrm{SC}}\ {}_{\mathrm{C}}\mathbf{p} + {}_{\mathrm{S}}\mathbf{r}_{\mathrm{SC}} \tag{4.19}$$

$$\frac{\partial \mathbf{h}}{\partial_{\mathrm{S}}\mathbf{p}} = \frac{\partial \mathbf{h}}{\partial_{\mathrm{C}}\mathbf{p}} \frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{S}}\mathbf{p}} \tag{4.20}$$

$$\frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{S}}\mathbf{p}} = \mathbf{C}_{\mathrm{CS}} \tag{4.21}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{T}_{\mathrm{SC}}} = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \delta\boldsymbol{\theta}} & \frac{\partial \mathbf{h}}{\partial_{\mathrm{S}}\mathbf{r}_{\mathrm{SC}}} \end{bmatrix} \tag{4.22}$$

$$\frac{\partial \mathbf{h}}{\partial \delta\boldsymbol{\theta}} = \frac{\partial \mathbf{h}}{\partial_{\mathrm{C}}\mathbf{p}} \frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{S}}\mathbf{p}} \frac{\partial_{\mathrm{S}}\mathbf{p}}{\partial \delta\boldsymbol{\theta}} \tag{4.23}$$

$$\frac{\partial_{\mathrm{S}}\mathbf{p}}{\partial \delta\boldsymbol{\theta}} = -\lfloor \mathbf{C}\{\delta\boldsymbol{\theta}\}\ {}_{\mathrm{S}}\mathbf{p}\ \times \rfloor \tag{4.24}$$

$$\frac{\partial \mathbf{h}}{\partial_{\mathrm{S}}\mathbf{r}_{\mathrm{SC}}} = \frac{\partial \mathbf{h}}{\partial_{\mathrm{C}}\mathbf{p}} \frac{\partial_{\mathrm{C}}\mathbf{p}}{\partial_{\mathrm{S}}\mathbf{p}} \frac{\partial_{\mathrm{S}}\mathbf{p}}{\partial_{\mathrm{S}}\mathbf{r}_{\mathrm{SC}}} \tag{4.25}$$

$$\frac{\partial_{\mathrm{S}}\mathbf{p}}{\partial_{\mathrm{S}}\mathbf{r}_{\mathrm{SC}}} = \mathbf{I} \tag{4.26}$$

$$\tag{4.27}$$

4.3 Jacobian w.r.t Fiducial Pose, $\mathbf{T_{WF}}$

$$\begin{aligned} {}_{\mathrm{W}}\mathbf{p} &= \mathbf{T}_{\mathrm{WF}} \ {}_{\mathrm{F}}\mathbf{p} \\ &= \mathbf{C}_{\mathrm{WF}} \ {}_{\mathrm{F}}\mathbf{p} + {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WF}} \end{aligned} \tag{4.28}$$

$$\frac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{p}} = \frac{\partial \mathbf{h}}{\partial {}_{\mathrm{C}}\mathbf{p}} \ \frac{\partial {}_{\mathrm{C}}\mathbf{p}}{\partial {}_{\mathrm{W}}\mathbf{p}} \tag{4.29}$$

$$\frac{\partial {}_{\mathrm{C}}\mathbf{p}}{\partial {}_{\mathrm{W}}\mathbf{p}} = \mathbf{C}_{\mathrm{CW}} \tag{4.30}$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{T}_{\mathrm{WF}}} = \left[\frac{\partial \mathbf{h}}{\partial \delta\boldsymbol{\theta}} \ \frac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WF}}} \right] \tag{4.31}$$

$$\frac{\partial \mathbf{h}}{\partial \delta\boldsymbol{\theta}} = \frac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{p}} \ \frac{\partial {}_{\mathrm{W}}\mathbf{p}}{\partial \delta\boldsymbol{\theta}} \tag{4.32}$$

$$\frac{\partial {}_{\mathrm{W}}\mathbf{p}}{\partial \delta\boldsymbol{\theta}} = -\lfloor \mathbf{C}\{\delta\boldsymbol{\theta}\} \ {}_{\mathrm{F}}\mathbf{p} \ \times \rfloor \tag{4.33}$$

$$\frac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WF}}} = \frac{\partial \mathbf{h}}{\partial {}_{\mathrm{W}}\mathbf{p}} \ \frac{\partial {}_{\mathrm{W}}\mathbf{p}}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WF}}} \tag{4.34}$$

$$\frac{\partial {}_{\mathrm{W}}\mathbf{p}}{\partial {}_{\mathrm{W}}\mathbf{r}_{\mathrm{WF}}} = \mathbf{I} \tag{4.35}$$

Bibliography

[1] P. Furgale, "Representing Robot Pose: The good, the bad, and the ugly." `http://paulfurgale.info/news/2014/6/9/representing-robot-pose-the-good-the-bad-and-the-ugly`, 2014. Accessed: 2018-11-15.

[2] J. Solà, "Quaternion kinematics for the error-state Kalman filter," Nov. 2017.

[3] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.