VIRTUAL CAMERA TUTORIAL

Brian O'Sullivan.

Palau de Congressos, Barcelona.

8th October / 2024

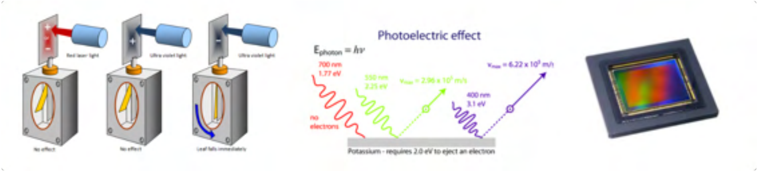
AutoSens

Introduction and Motivation

Overview

All information presented in these slides is freely available online, or can be readily deduced with known logic and mathematics. All views and options shared are entirely my own.

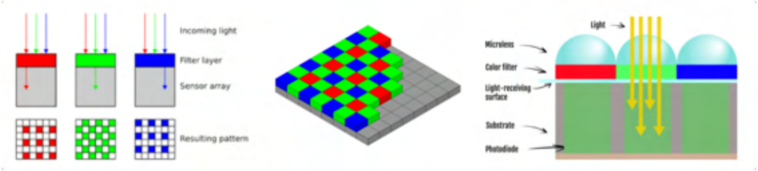
Introduction and Motivation



The PhotoElectric Effect

- Discovered in 1887 by Heinrich Hertz, who observed that UV light changes the voltage between two metal electrodes.
- Established there exists a relationship between light and electricity.
- Albert Einstein explained that light comes in packets of energy 'photons'. When a photon's energy is greater than the threshold energy, an electron is released.

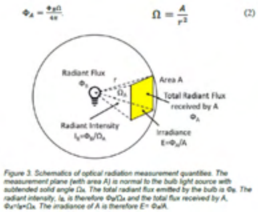
Introduction and Motivation

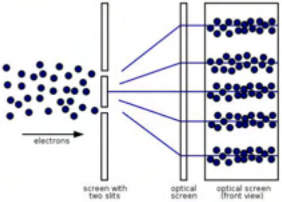


The PhotoElectric Effect is a non-reversable process

- We do not know the wavelength of light that created the electron, only that it was above a certain threshold energy.
- The Bayer Colour Filter Array is used to recover colour (wavelength) information about the incident light.
- Image Signal Processors converts the raw RGB data using various image quality enhancing techniques to produce an appealing digital photograph.

Introduction and Motivation

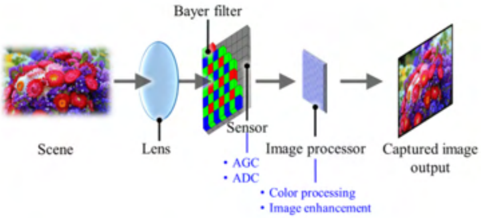




Wave function collapse in the photo electric effect:
Light extends from a source as a spherical wavefront, to collapse and transfer it’s energy to an electron.

- Light is composed of packets of energy called photons that behave matter-like.
- Matter can be demonstrated to exhibit wave properties, previously thought to be exclusive to light.

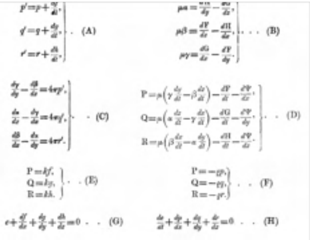
Introduction and Motivation



The ElectroMagnetic Field

- Light interacts with photo diodes and Lens to produce data.
- Processed data produces a representative image of the world.
- Lens design is based on accurate simulation of the EM-Field.
- Image Quality relies on accurate simulation and correction of light-matter interactions.

Introduction and Motivation



James Clerk Maxwell’s original Treatise of the Electromagnetic Field (1864), involved 20 partial differential equations to solve for 20 unknowns of the Electromagnetic Field.
In 1873, he published a two-volume work "A Treatise on Electricity and Magnetism". Maxwell used Quaternions to simplify the field equations, and ended up with 11 vectors (33 symbols), and that was how the math was left since his passing in 1879.

Introduction and Motivation

Oliver Heaviside

Name	Differential form	Integral form
Gauss's law	$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$	$\oiint_{\partial V} \mathbf{E} \cdot d\mathbf{A} = \frac{Q(V)}{\varepsilon_0}$
Gauss's law for magnetism	$\nabla \cdot \mathbf{B} = 0$	$\oiint_{\partial V} \mathbf{B} \cdot d\mathbf{A} = 0$
Maxwell–Faraday equation (Faraday's law of induction)	$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$	$\oint_{\partial S} \mathbf{E} \cdot d\mathbf{l} = -\frac{\partial \Phi_{B,S}}{\partial t}$
Ampère's circuital law (with Maxwell's correction)	$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}$	$\oint_{\partial S} \mathbf{B} \cdot d\mathbf{l} = \mu_0 I_S + \mu_0 \varepsilon_0 \frac{\partial \Phi_{E,S}}{\partial t}$

Maxwell's Equations

The equations of the electromagnetic field, as we know them today, were formulated by Oliver Heaviside, based on Maxwell's Treatise.

Heaviside's contribution represents the latest development in simplifying Maxwell's Field equations, where 12 of Maxwell's equations were reduced to 4, referred to as Gauss's Law, Gauss's Law of Magnetism, Faraday's Law of Induction and Ampere's Law.

Introduction and Motivation

Oliver Heaviside

Name	Differential form	Integral form
Gauss's law	$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$	$\oiint_{\partial V} \mathbf{E} \cdot \mathrm{d}\mathbf{A} = \frac{Q(V)}{\varepsilon_0}$
Gauss's law for magnetism	$\nabla \cdot \mathbf{B} = 0$	$\oiint_{\partial V} \mathbf{B} \cdot \mathrm{d}\mathbf{A} = 0$
Maxwell–Faraday equation (Faraday's law of induction)	$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$	$\oint_{\partial S} \mathbf{E} \cdot \mathrm{d}\mathbf{l} = -\frac{\partial \Phi_{B,S}}{\partial t}$
Ampère's circuital law (with Maxwell's correction)	$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}$	$\oint_{\partial S} \mathbf{B} \cdot \mathrm{d}\mathbf{l} = \mu_0 I_S + \mu_0 \varepsilon_0 \frac{\partial \Phi_{E,S}}{\partial t}$

Maxwell's Equations

'on proceeding to apply quaternionics to the development of electrical theory, I found it very inconvenient. Quaternionics was in its vectorial aspects antiphysical and unnatural, and did not harmonise with common scalar mathematics. So I dropped out the quaternion altogether, and kept to pure scalar and vectors.'

Oliver Heaviside, Electromagnetic Theory (1912), Volume III; Appendix K: Vector Analysis, p. 136; "The Electrician" Pub. Co., London.

Introduction and Motivation

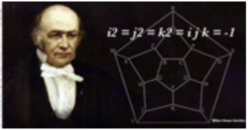
Quaternions were discovered by William Rowan Hamilton in 1843.
- To describe 3 dimensional rotations as an algebra.

"You know that I have long wished, ...to possess a Theory of Triplets,... I discovered yesterday a theory of quaternions which includes such a theory of triplets...

And here there dawned on me the notion that we must admit, in some sense, a fourth dimension of space for the purpose of calculating with triples."

Extract from a letter dated October 17^{th}, 1843 from Hamilton to John T. Graves.

Introduction and Motivation



Important precursors to the development of quaternions were Euler's four-square identity (1748) and Olinde Rodrigues' parameterization of general rotations by four parameters (1840).

Quaternions are essential for all applications requiring descriptions of 3D rotations, from aeronautics, to computer graphics, robotics, to gyroscopes, and the general theory of Classical Mechanics.

In the context of the current presentation, quaternions are essential for describing the extrinsics of camera calibration.

Hamilton's new System of Imaginaries in Algebra

`Quaternions`

Hamilton's new System of Imaginaries in Algebra

In this presentation of the quaternions we begin from Hamilton's original treatise, *On Quaternions; or on a new System of Imaginaries in Algebra*, by Sir William Rowan Hamilton, which appeared in 18 instalments in volumes xxv-xxxvi of The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science (3rd Series), for the years 1844-1850.

Thereafter we develop the theory of quaternions to give a comprehensive account of the known theory, for an engineering perspective.

Hamilton's new System of Imaginaries in Algebra

Let an expression of the form:

$$Q = w + ix + jy + kz$$

be called a quaternion, when w, x, y, z, which we shall call the four constituents of the quaternion Q, denote any real quantities, positive or negative or null, but i, j, k are symbols of three imaginary quantities, which we shall call imaginary units.

Hamilton's new System of Imaginaries in Algebra

It will then be natural to define that the addition or subtraction of quaternions is effected by the formula:

$$Q \pm Q' = (w \pm w') + i(x \pm x') + j(y \pm y') + k(z \pm z')$$

or, in words, by the rule, that the sums or differences of the constituents of any two quaternions, are the constituents of the sum or difference of those two quaternions themselves.

Hamilton's new System of Imaginaries in Algebra

It will also be natural to define that the product QQ', of the multiplication of Q as a multiplier into Q' as a multiplicand, is capable of being thus expressed:

$$\begin{aligned} QQ' &= ww' + iwx' + jwy' + kwz' \\ &\quad + ixw' + i^2xx' + ijxy' + ikxz' \\ &\quad + jyw' + jiyx' + j^2yy' + jkyz' \\ &\quad + kzw' + kizx' + kjzy' + k^2zz' \end{aligned}$$

Hamilton's new System of Imaginaries in Algebra

`but before we can reduce this product to an expression of the quaternion form, such as`

$$QQ' = Q = w'' + ix'' + jy'' + kz''$$

`it is necessary to fix on quaternion-expressions (or on real values) for the nine squares or products,`

Hamilton's new System of Imaginaries in Algebra

Considerations, which it might occupy too much space to give an account of on the present occasion, have led the writer to adopt the following system of values or expressions for these nine squares or products:

$$i^2 = j^2 = k^2 = -1 \tag{A.}$$

$$ij = k, \quad jk = i, \quad ki = j; \tag{B.}$$

$$ji = -k \quad kj = -i, \quad ik = -j \tag{C.}$$

though it must, at first sight, seem strange and almost unallowable, to define that the product of two imaginary factors in one order differs (in sign) from the product of the same factors in the opposite order. $(ji = -ij)$

Hamilton's new System of Imaginaries in Algebra

With the assumed relations (A.), (B.), (C.), we have the four following expressions for the four constituents of the product of two quaternions, as functions of the constituents of the multiplier and multiplicand:

$$\begin{aligned} w'' &= ww' - xx' - yy' - zz', \\ x'' &= wx' + xw' + yz' - zy', \\ y'' &= wy' + yw' + zx' - xz', \\ z'' &= wz' + zw' + xy' - yx', \end{aligned}$$

These equations give

$$w''^2 + x''^2 + y''^2 + z''^2 = (w^2 + x^2 + y^2 + z^2)(w'^2 + x'^2 + y'^2 + z'^2)$$

Hamilton's new System of Imaginaries in Algebra

and therefore $\mu'' = \mu\mu'$

if we introduce a system of expressions for the constituents, of the forms

$$
\begin{aligned}
w &= \mu\cos(\tfrac{\beta}{2}) \\
x &= \mu\sin(\tfrac{\beta}{2})\sin(\theta)\cos(\phi), \\
y &= \mu\sin(\tfrac{\beta}{2})\sin(\theta)\sin(\phi), \\
z &= \mu\sin(\tfrac{\beta}{2})\cos(\theta),
\end{aligned}
$$

and suppose each μ to be positive. Calling, therefore, μ the modulus of the quaternion Q, we have this theorem: that the modulus of the product Q'' of any two quaternions Q and Q', is equal to the product of their moduli.

---

co-ordinate system convention is changed to suit the current presentation.

Hamilton's new System of Imaginaries in Algebra

Here and in the following we consider only unit quaternions, such that $\mu = 1$. The (i, j, k) vector describes the 2-sphere coordinates.

$$\vec{r} = \begin{pmatrix} \sin(\theta)\cos(\phi), \\ \sin(\theta)\sin(\phi), \\ \cos(\theta), \end{pmatrix} \tag{1}$$

let this point $\vec{r}$ be called the representative point of the quaternion Q. Let $\vec{r}'$ and $\vec{r}''$ be, in like manner, the representative points of Q' and Q''; ...in the spherical triangle $\vec{r}\,\vec{r}'\,\vec{r}''$, formed by the representative points of the two factors and the product (in any multiplication of two quaternions)...

$$Q'' = Q\,Q'$$

The spherical triangle, and scalar-vector decomposition

The spherical triangle and scalar-vector decomposition of the quaternion

The spherical triangle, and scalar-vector decomposition

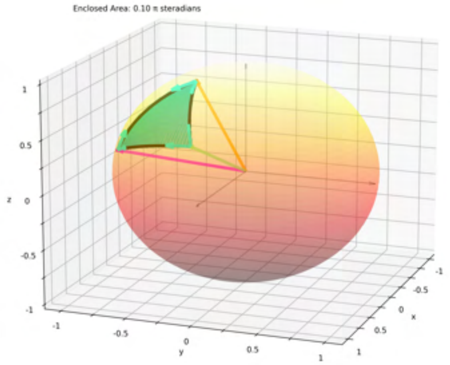
The enclosed area of the spherical triangle is given by:

$$Area = \angle A + \angle B + \angle C - \pi$$

where the interior angles are the inverse cosine of the respective unit tangent vectors:

$$\angle A = \cos^{-1}\left(\vec{t}_{AB} \cdot \vec{t}_{AC}\right)$$
$$\angle B = \cos^{-1}\left(\vec{t}_{BA} \cdot \vec{t}_{BC}\right)$$
$$\angle C = \cos^{-1}\left(\vec{t}_{CA} \cdot \vec{t}_{CB}\right)$$

→ the enclosed area is measured in Steradians.



The spherical triangle $\vec{r}\ \vec{r}'\ \vec{r}''$, formed by the representative points of quaternions Q, Q', Q''.

The spherical triangle, and scalar-vector decomposition

Consider

$$\vec{Q} = \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\frac{\beta}{2}) \\ \sin(\frac{\beta}{2})\sin(\theta)\cos(\phi) \\ \sin(\frac{\beta}{2})\sin(\theta)\sin(\phi) \\ \sin(\frac{\beta}{2})\cos(\theta) \end{pmatrix} = \begin{pmatrix} \cos(\frac{\beta}{2}) \\ \sin(\frac{\beta}{2})\vec{r} \end{pmatrix} = \begin{pmatrix} q_1 \\ \vec{q} \end{pmatrix} = \begin{pmatrix} q_1 \\ q_i \\ q_j \\ q_k \end{pmatrix}$$

The quaternion product can be decomposed as

$$\vec{Q}\vec{Q}' = \begin{pmatrix} q_1 q_1' - \vec{q}\cdot\vec{q}' \\ q_1\vec{q}' + q_1'\vec{q} + \vec{q}\times\vec{q}' \end{pmatrix} = \begin{pmatrix} q_1 q_1' - q_i q_i' - q_j q_j' - q_k q_k', \\ q_1 q_i' + q_i q_1' + q_j q_k' - q_k q_j', \\ q_1 q_j' + q_j q_1' + q_k q_i' - q_i q_k', \\ q_1 q_k' + q_k q_1' + q_i q_j' - q_j q_i', \end{pmatrix}$$

The spherical triangle, and scalar-vector decomposition

Quaternion multiplication is non-commutative, $\vec{Q}\vec{Q}' \neq \vec{Q}'\vec{Q}$.

$$\vec{Q}\vec{Q}' = \begin{pmatrix} q_1 q_1' - \vec{q}\cdot\vec{q}' \\ q_1\vec{q}' + q_1'\vec{q} + \vec{q}\times\vec{q}' \end{pmatrix}$$

$$\vec{Q}'\vec{Q} = \begin{pmatrix} q_1' q_1 - \vec{q}'\cdot\vec{q} \\ q_1'\vec{q} + q_1\vec{q}' + \vec{q}'\times\vec{q} \end{pmatrix}$$

Therefore

$$\vec{Q}\vec{Q}' - \vec{Q}'\vec{Q} = \begin{pmatrix} 0 \\ 2\vec{q}\times\vec{q}' \end{pmatrix}$$

since $\vec{q}\times\vec{q}' = -\vec{q}'\times\vec{q}$

The spherical triangle, and scalar-vector decomposition

The conjugate quaternion

$$\vec{Q}^* = \begin{pmatrix} q_1 \\ -\vec{q} \end{pmatrix}$$

Setting $\vec{Q}' = \vec{Q}^*$, the quaternion product is

$$\vec{Q}\vec{Q}^* = \begin{pmatrix} q_1 q_1 - \vec{q}\cdot(-\vec{q}) \\ \cancel{q_1(-\vec{q}) + q_1\vec{q}} + \cancel{\vec{q}\times(-\vec{q})} \end{pmatrix} = \begin{pmatrix} q_1 q_1 + \vec{q}\cdot\vec{q} \\ \vec{0} \end{pmatrix} = \begin{pmatrix} 1 \\ \vec{0} \end{pmatrix}$$

since $\mu = \sqrt{q_1^2 + q_i^2 + q_j^2 + q_k^2} = 1$.

The unit quaternion is an element of the 3-sphere, $\mathbb{S}^3$, embedded in the 4-dimensional Euclidian space $\mathbb{R}^4$.

$$\vec{Q} \in \mathbb{S}^3 \subset \mathbb{R}^4$$

The spherical triangle, and scalar-vector decomposition

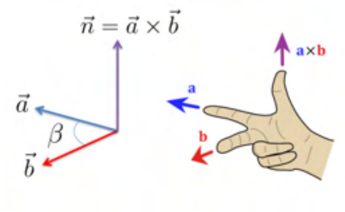
The unit 3-vector $\vec{a}$ is an element of the 2-sphere,

$$\vec{a} \in \mathbb{S}^2 \subset \mathbb{R}^3$$

The conjugate product of a unit quaternion $\vec{Q}$ with $\vec{a}$,to the new position $\vec{b}$, is denoted

$$\vec{b} = \vec{Q}\,\vec{a}\,\vec{Q}^*$$

The conjugate product represents the rotation of $\vec{a}$ around the axis $\vec{n} = \vec{a} \times \vec{b} / |\vec{a} \times \vec{b}|$, by an angle β.



The spherical triangle, and scalar-vector decomposition

Problem statement: Given the vectors $\vec{a}$, $\vec{b} \in \mathbb{S}^2$ find the quaternion $\vec{Q}$ that describes the rotation.

Solution:

$$\vec{n} = \frac{\vec{a} \times \vec{b}}{|\vec{a} \times \vec{b}|} \qquad \beta = 2 * \cos^{-1}\left(\vec{a} \cdot \vec{b}\right)$$

$$\vec{Q} = \begin{pmatrix} \cos\left(\frac{\beta}{2}\right) \\ \vec{n}\sin\left(\frac{\beta}{2}\right) \end{pmatrix} = \begin{pmatrix} q_1 \\ \vec{q} \end{pmatrix}$$

developing [since $\vec{a} \perp \vec{q}$, then $\vec{a} \cdot \vec{q} = 0$]

$$\vec{b} = \vec{Q}\,\vec{a}\,\vec{Q}^* = \begin{pmatrix} q_1 \\ \vec{q} \end{pmatrix} \begin{pmatrix} -\vec{a} \cdot \vec{q} \\ q_1 \vec{a} - \vec{a} \times \vec{q} \end{pmatrix} = \begin{pmatrix} q_1 \\ \vec{q} \end{pmatrix} \begin{pmatrix} 0 \\ q_1 \vec{a} - \vec{a} \times \vec{q} \end{pmatrix}$$

$$\vec{b} = \begin{pmatrix} -\vec{q} \cdot (q_1 \vec{a} - \vec{a} \times \vec{q}) \\ q_1 (q_1 \vec{a} - \vec{a} \times \vec{q}) + \vec{q} \times (q_1 \vec{a} - \vec{a} \times \vec{q}) \end{pmatrix}$$

The spherical triangle, and scalar-vector decomposition

Since $\vec{a}\cdot\left(\vec{a}\times\vec{q}\right)=\vec{q}\cdot\left(\vec{a}\times\vec{q}\right)=0.$

$$\vec{b} = \begin{pmatrix} 0 \\ q_1^2\vec{a}-\vec{q}\times\left(\vec{a}\times\vec{q}\right)-2q_1\left(\vec{a}\times\vec{q}\right) \end{pmatrix}$$

Vector triple product identity, $\vec{A}\times\left(\vec{B}\times\vec{C}\right)=\left(\vec{A}\cdot\vec{C}\right)\vec{B}-\left(\vec{A}\cdot\vec{B}\right)\vec{C}$

$$\rightarrow\ \vec{q}\times\left(\vec{a}\times\vec{q}\right) = \left(\vec{q}\cdot\vec{q}\right)\vec{a}$$

Plugging in

$$\vec{b} = \left(q_1^2-\vec{q}\cdot\vec{q}\right)\vec{a}+2q_1\left(\vec{q}\times\vec{a}\right)$$

The spherical triangle, and scalar-vector decomposition

Plugging in, $q_1 = \cos\left(\frac{\beta}{2}\right)$, and $\vec{q} = \sin\left(\frac{\beta}{2}\right)\vec{n}$,

$$\vec{b} = \left(\cos^2\left(\frac{\beta}{2}\right) - \sin^2\left(\frac{\beta}{2}\right)\right)\vec{a} + 2\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\beta}{2}\right)(\vec{n}\times\vec{a})$$

Trigonometric identities

$$\begin{aligned}\cos(\beta) &= \cos^2\left(\frac{\beta}{2}\right) - \sin^2\left(\frac{\beta}{2}\right)\\ \sin(\beta) &= 2\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\beta}{2}\right)\end{aligned}$$

Therefore

$$\vec{b}_a = \cos(\beta)\,\vec{a} + \sin(\beta)(\vec{n}\times\vec{a})$$

and through an analogous construction

$$\vec{a}_b = \cos(\beta)\,\vec{b} - \sin(\beta)\left(\vec{n}\times\vec{b}\right)$$

The spherical triangle, and scalar-vector decomposition

The unit tangent vector $\vec{t}_{AB}$ along the arc from $\vec{a}$ to $\vec{b}$, is minus the (normalized) partial derivative with respect to β of the rotated vector $\vec{a}_b$, from $\vec{b}$. By analogy we find the unit tangent vector $\vec{t}_{BA}$. With

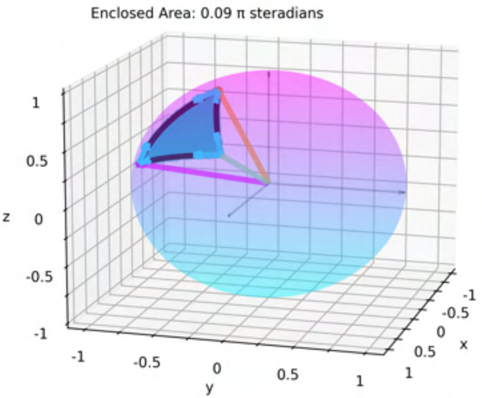
$$\begin{aligned} \vec{a}_b &= \cos(\beta)\,\vec{b} - \sin(\beta)\left(\vec{n}\times\vec{b}\right) & \partial_\beta \vec{a}_b &= -\sin(\beta)\,\vec{b} - \cos(\beta)\left(\vec{n}\times\vec{b}\right) \\ \vec{b}_a &= \cos(\beta)\,\vec{a} + \sin(\beta)\,(\vec{n}\times\vec{a}) & \partial_\beta \vec{b}_a &= -\sin(\beta)\,\vec{a} + \cos(\beta)\,(\vec{n}\times\vec{a}) \end{aligned}$$

We have

$$\vec{t}_{AB} = \frac{-\partial_\beta \vec{a}_b}{\sqrt{\partial_\beta \vec{a}_b \cdot \partial_\beta \vec{a}_b}} \qquad \vec{t}_{BA} = \frac{-\partial_\beta \vec{b}_a}{\sqrt{\partial_\beta \vec{b}_a \cdot \partial_\beta \vec{b}_a}}$$

Consequently we can calculate the 3 angles of the spherical triangle $\angle A, \angle B, \angle C$, and determine the enclosed area in steradians.

The spherical triangle, and scalar-vector decomposition



Python scripts available at github.com/mo-geometry

SU(2): Special Unitary Group of 2x2 complex matrices

```
SU(2):  Special Unitary Group of 2x2 complex matrices
```

SU(2): Special Unitary Group of 2x2 complex matrices

SU(2) is the group of 2x2 complex matrices of the form

$$\hat{U} = \begin{pmatrix} a+ib & c+id \\ -c+id & a-ib \end{pmatrix}$$

for $a, b, c, d \in \mathbb{R}$, with unit determinant $a^2 + b^2 + c^2 + d^2 = 1$.
The quaternions are isomorphic to the SU(2) group.
The Cayley matrices are the natural basis of the SU(2) quaternion

$$\hat{\sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \hat{\sigma}_i = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} \quad \hat{\sigma}_j = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \hat{\sigma}_k = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}$$

which satisfy

$$\hat{\sigma}_i^2 = \hat{\sigma}_j^2 = \hat{\sigma}_k^2 = \hat{\sigma}_i\hat{\sigma}_j\hat{\sigma}_k = -\hat{\sigma}_1$$

SU(2): Special Unitary Group of 2x2 complex matrices

The quaternion, and it's conjugate, is expanded in the SU(2) basis

$$\begin{aligned} \hat{U} &= a\,\hat{\sigma}_1 + b\,\hat{\sigma}_i + c\,\hat{\sigma}_j + d\,\hat{\sigma}_k \\ \hat{U}^\dagger &= a\,\hat{\sigma}_1 - b\,\hat{\sigma}_i - c\,\hat{\sigma}_j - d\,\hat{\sigma}_k \end{aligned}$$

$\hat{U}\hat{U}^\dagger = \hat{\sigma}_1$, where $\dagger$ refers to the transpose conjugate, and $\hat{\sigma}_\bullet^\dagger = -\hat{\sigma}_\bullet$, for $\bullet = i, j, k$.
The 3-vector $\hat{r}$ is expanded as

$$\hat{r} = r_i\hat{\sigma}_i + r_j\hat{\sigma}_j + r_k\hat{\sigma}_k$$

and is rotated to the position $\hat{r}'$ via

$$\hat{r}' = \hat{U}\,\hat{r}\,\hat{U}^\dagger$$

SU(2): Special Unitary Group of 2x2 complex matrices

Dynamics of Time Dependent systems:

The 3-vector evolves in time from an initial state $\hat{r}_0$ via the time dependent quaternion $\hat{U}(t)$, as

$$\hat{r}(t) = \hat{U}(t)\,\hat{r}_0\,\hat{U}^\dagger(t)$$

We develop the first derivative as

$$\begin{aligned}
\dot{\hat{r}} &= \dot{\hat{U}}\,\hat{r}_0\,\hat{U}^\dagger + \hat{U}\,\hat{r}_0\,\dot{\hat{U}}^\dagger \\
\dot{\hat{r}} &= \dot{\hat{U}}\,\hat{U}^\dagger\left(\hat{U}\,\hat{r}_0\,\hat{U}^\dagger\right) + \left(\hat{U}\,\hat{r}_0\,\hat{U}^\dagger\right)\hat{U}\,\dot{\hat{U}}^\dagger \\
\dot{\hat{r}} &= \dot{\hat{U}}\,\hat{U}^\dagger\,\hat{r} + \hat{r}\,\hat{U}\,\dot{\hat{U}}^\dagger
\end{aligned}$$

Define the Hamiltonian

$$\hat{\mathscr{H}} \equiv \dot{\hat{U}}\,\hat{U}^\dagger$$

SU(2): Special Unitary Group of 2x2 complex matrices

Since $\hat{U}\hat{U}^\dagger = \hat{\sigma}_1$, then $\hat{U}\dot{\hat{U}}^\dagger = -\dot{\hat{U}}\hat{U}^\dagger$.

$$\dot{\hat{r}} = \hat{\mathscr{H}}\hat{r} - \hat{r}\hat{\mathscr{H}}$$

We arrive at the Von Neumann equation of motion,

$$\dot{\hat{r}} = \left[\hat{\mathscr{H}}, \hat{r}\right]$$

The elements of the Hamiltonian

$$\hat{\mathscr{H}} = \frac{\mathscr{H}_i}{2}\hat{\sigma}_i + \frac{\mathscr{H}_j}{2}\hat{\sigma}_j + \frac{\mathscr{H}_k}{2}\hat{\sigma}_k$$

are given by:

$$\begin{aligned} \mathscr{H}_i &= 2\left(c\dot{d} - \dot{c}d\right) + 2\left(a\dot{b} - \dot{a}b\right) \\ \mathscr{H}_j &= 2\left(\dot{b}d - b\dot{d}\right) + 2\left(a\dot{c} - \dot{a}c\right) \\ \mathscr{H}_k &= 2\left(b\dot{c} - \dot{b}c\right) + 2\left(a\dot{d} - \dot{a}d\right) \end{aligned}$$

SU(2): Special Unitary Group of 2x2 complex matrices

Alternatively, we need consider

$$\hat{r}'(t) = \hat{U}^{\dagger}(t)\,\hat{r}'_0\,\hat{U}(t)$$

Developing in a similar manner we arrive at a slightly different version of the Von Neumann equation of motion,

$$\dot{\hat{r}}' = \left[\hat{\mathscr{H}}^{-}, \hat{r}'\right]$$

where

$$\begin{aligned}
\mathscr{H}_i^{-} &= 2\left(c\dot{d} - \dot{c}d\right) - 2\left(a\dot{b} - \dot{a}b\right) \\
\mathscr{H}_j^{-} &= 2\left(\dot{b}d - b\dot{d}\right) - 2\left(a\dot{c} - \dot{a}c\right) \\
\mathscr{H}_k^{-} &= 2\left(b\dot{c} - \dot{b}c\right) - 2\left(a\dot{d} - \dot{a}d\right)
\end{aligned}$$

SU(2): Special Unitary Group of 2x2 complex matrices

By construction the general form of the Von Neumann equation of motion is,

$$\dot{\hat{r}} = \left[\hat{\mathscr{H}}^{\pm}, \hat{r}\right]$$

where the elements of the Hamiltonian are respectively

$$\mathscr{H}^{\pm}_{\bullet} = \mathscr{B}_{\bullet} \pm \mathscr{E}_{\bullet}$$

for $\bullet = i, j, k$ and

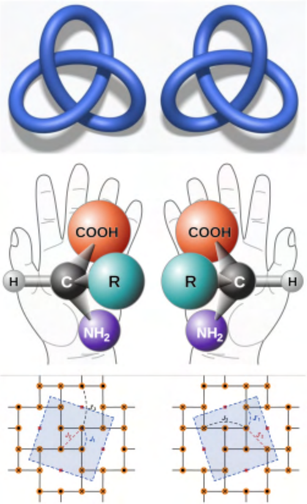
$$\begin{aligned} \mathscr{B}_i &= 2\left(c\dot{d} - \dot{c}d\right) & \qquad \mathscr{E}_i &= 2\left(a\dot{b} - \dot{a}b\right) \\ \mathscr{B}_j &= 2\left(\dot{b}d - b\dot{d}\right) & \qquad \mathscr{E}_j &= 2\left(a\dot{c} - \dot{a}c\right) \\ \mathscr{B}_k &= 2\left(b\dot{c} - \dot{b}c\right) & \qquad \mathscr{E}_k &= 2\left(a\dot{d} - \dot{a}d\right) \end{aligned}$$

Chirality and the Electromagnetic Field Tensor

`Chirality and the Electromagnetic Field Tensor`

Chirality and the Electromagnetic Field Tensor

In geometry, Chirality refers to the characteristic of an object that cannot be superimposed on its mirror image, by rotations and translations alone.



Chiral Quaternions

Represented as 4 dimensional matrices, the quaternion has a pair of Chiral representations.
In the following these are referred to as the Left Cayley matrices, and Right Cayley matrices respectively.

Chirality and the Electromagnetic Field Tensor

The Left Cayley matrices:

$$\hat{l}_i = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \hat{l}_j = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \quad \hat{l}_k = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

satisfies the quaternion identity

$$\hat{l}_i^2 = \hat{l}_j^2 = \hat{l}_k^2 = \hat{l}_i \hat{l}_j \hat{l}_k = -\hat{\mathbb{1}}$$

Consequently the Left Cayley Quaternion is expressed

$$\hat{U}_L = a\,\hat{\mathbb{1}} + b\,\hat{l}_i + c\,\hat{l}_j + d\,\hat{l}_k$$

with

$$\hat{U}_L \hat{U}_L^t = \hat{\mathbb{1}}$$

Chirality and the Electromagnetic Field Tensor

The Right Cayley matrices:

$$\hat{r}_i = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \hat{r}_j = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \quad \hat{r}_k = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}$$

satisfies the quaternion identity

$$\hat{r}_i^2 = \hat{r}_j^2 = \hat{r}_k^2 = \hat{r}_i \hat{r}_j \hat{r}_k = -\hat{\mathbb{1}}$$

Consequently the Right Cayley Quaternion is expressed

$$\hat{U}_R = a\,\hat{\mathbb{1}} + b\,\hat{r}_i + c\,\hat{r}_j + d\,\hat{r}_k$$

with

$$\hat{U}_R \hat{U}_R^t = \hat{\mathbb{1}}$$

Chirality and the Electromagnetic Field Tensor

The Left and Right Cayley quaternions:

$$\hat{U}_L = \begin{pmatrix} a & -b & -c & -d \\ b & a & -d & -c \\ c & d & a & -b \\ d & c & b & a \end{pmatrix} \qquad \hat{U}_R = \begin{pmatrix} a & b & c & d \\ -b & a & -d & -c \\ -c & d & a & -b \\ -d & c & b & a \end{pmatrix}$$

$\hat{U}_L, \hat{U}_R$, are commutative, such that

$$[\hat{U}_L, \hat{U}_R^t] = [\hat{U}_L^t, \hat{U}_R] = [\hat{U}_L, \hat{U}_R] = [\hat{U}_L^t, \hat{U}_R^t] = \hat{0}$$

Chirality and the Electromagnetic Field Tensor

- Classical Electrodynamics is described by the products

$$\hat{U}_L \hat{U}_R^t \qquad \text{and} \qquad \hat{U}_L^t \hat{U}_R$$

- Classical Mechanics is described by the products

$$\hat{U}_L \hat{U}_R \qquad \text{and} \qquad \hat{U}_L^t \hat{U}_R^t$$

Chirality and the Electromagnetic Field Tensor

Classical Electrodynamics:

$$\hat{F} = \hat{U}_L \hat{U}_R^t = \begin{pmatrix} \alpha_1 & -2ab & -2ac & -2ad \\ 2ab & \alpha_2 & -2bc & -2bd \\ 2ac & -2bc & \alpha_3 & -2cd \\ 2ad & -2bd & -2cd & \alpha_4 \end{pmatrix}$$

with the diagonal elements given by

$$\begin{aligned} \alpha_1 &= a^2 - b^2 - c^2 - d^2 \\ \alpha_2 &= a^2 - b^2 + c^2 + d^2 \\ \alpha_3 &= a^2 + b^2 - c^2 + d^2 \\ \alpha_4 &= a^2 + b^2 + c^2 - d^2 \end{aligned}$$

Chirality and the Electromagnetic Field Tensor

Classical Electrodynamics:

The Electromagnetic Field Tensor is defined:

Contravariant form

$$\mathscr{F}_{\mu\nu} = \dot{\hat{F}}\hat{F}^t = \begin{pmatrix} 0 & \mathscr{E}_i & \mathscr{E}_j & \mathscr{E}_k \\ -\mathscr{E}_i & 0 & -\mathscr{B}_k & \mathscr{B}_j \\ -\mathscr{E}_j & \mathscr{B}_k & 0 & -\mathscr{B}_i \\ -\mathscr{E}_k & -\mathscr{B}_j & \mathscr{B}_i & 0 \end{pmatrix}$$

Covariant form

$$\mathscr{F}^{\mu\nu} = \dot{\hat{F}}^t\hat{F} = \begin{pmatrix} 0 & -\mathscr{E}_i & -\mathscr{E}_j & -\mathscr{E}_k \\ \mathscr{E}_i & 0 & -\mathscr{B}_k & \mathscr{B}_j \\ \mathscr{E}_j & \mathscr{B}_k & 0 & -\mathscr{B}_i \\ \mathscr{E}_k & -\mathscr{B}_j & \mathscr{B}_i & 0 \end{pmatrix}$$

The Special Orthogonal Group of 3×3 matrices SO(3)

The Special Orthogonal Group of 3×3 matrices SO(3)

The Special Orthogonal Group of 3×3 matrices SO(3)

Classical Mechanics: The SO(3) rotation matrix is defined

$$\hat{R} = \hat{U}_L \hat{U}_R$$

$$\hat{R} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 0 & 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 0 & 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{pmatrix}$$

$\hat{R}\hat{R}^t = \hat{\mathbb{1}}.$

The Special Orthogonal Group of 3×3 matrices SO(3)

For brevity of vector operations we define:

$$\hat{R} \equiv \begin{pmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{pmatrix}$$

This is the general form of the Special Orthogonal Group of 3x3 matrices SO(3). The Hamiltonian's of Classical Mechanics are:

$$\hat{\mathscr{H}}^+ = \dot{\hat{R}}\hat{R}^t = \hat{\mathscr{B}} + \hat{\mathscr{E}} \qquad \hat{\mathscr{H}}^- = \dot{\hat{R}}^t\hat{R} = \hat{\mathscr{B}} - \hat{\mathscr{E}}$$

Therefore the 3-vector evolves in time according to

$$\dot{\vec{r}} = \hat{\mathscr{H}}^{\pm}\vec{r}$$

For the remainder we drop the ± notation and assume the +ive case.

The Special Orthogonal Group of 3×3 matrices SO(3)

Classical Mechanics: The Hamiltonian is expanded in the basis of the Lie Algebra matrices

$$\hat{\mathscr{H}} = \mathscr{H}_i \hat{\pi}_i + \mathscr{H}_j \hat{\pi}_j + \mathscr{H}_k \hat{\pi}_k$$

where

$$\hat{\pi}_i = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \hat{\pi}_j = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \qquad \hat{\pi}_k = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and in matrix form

$$\hat{\mathscr{H}} = \begin{pmatrix} 0 & -\mathscr{H}_k & \mathscr{H}_j \\ \mathscr{H}_k & 0 & -\mathscr{H}_i \\ -\mathscr{H}_j & \mathscr{H}_i & 0 \end{pmatrix}$$

The Special Orthogonal Group of 3×3 matrices SO(3)

Classical Mechanics:
The classical equation of motion is equivalently expressed as the vector cross product

$$\dot{\vec{r}} = \vec{\mathscr{H}} \times \vec{r}$$

From here the Classical analysis can be developed to account for moving frames, where the basis vectors of $\vec{r}$ are time dependent, i.e. $\left(\vec{\sigma}_i(t), \vec{\sigma}_j(t), \vec{\sigma}_k(t)\right)$. These are non-inertial frames of reference, and give rise to the so-called fictitious forces of Classical Mechanics.

- The Coriolis Force: $2\vec{\mathscr{H}} \times \dot{\vec{r}}$
- The Centrifugal Force: $\vec{\mathscr{H}} \times (\vec{\mathscr{H}} \times \vec{r})$
- The Euler Force: $\dot{\vec{\mathscr{H}}} \times \vec{r}$

Where the force is $F = m\ddot{\vec{r}}$, and the acceleration in the non-inertial frame is

$$\ddot{\vec{r}} = 2\vec{\mathscr{H}} \times \dot{\vec{r}} + \vec{\mathscr{H}} \times (\vec{\mathscr{H}} \times \vec{r}) + \dot{\vec{\mathscr{H}}} \times \vec{r}$$

Inertial Measurement Unit

Inertial Measurement Unit

Inertial Measurement Unit

Classical Mechanics: Example use case

1. Gyroscope: measures the angular rate of rotation **around** each body axes.

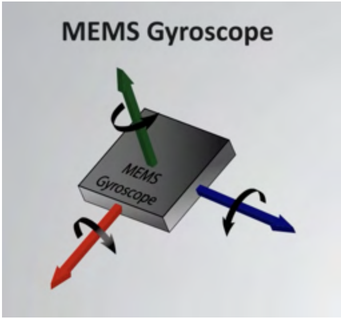
MEMS Gyroscopes read angular acceleration at very small time intervals dt. In the absence of noise and measurement drift, the orientation of the body under measurement updates as

$$\vec{r}(t_{n+1}) = \text{Exp}\left[\hat{\mathscr{H}}(t_n)dt\right]\vec{r}(t_n)$$

where the matrix exponential of the instantaneous Hamiltonian$*dt$

$$\hat{R}(t_n) = \text{Exp}\left[\hat{\mathscr{H}}(t_n)dt\right]$$

is the instantaneous rotation matrix.

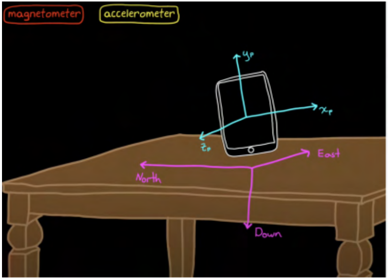


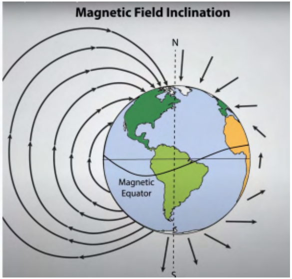
Inertial Measurement Unit

Classical Mechanics: Example use case

2. Magnetometer: measures the Earth's magnetic field, and local magnetic fields, **along** each body axes.
3. Accelerometer: measures the Earth's gravitational field, and local accelerations, **along** each body axes.

The absolute orientation of a body under measurement can be determined from the accelerometer and magnetometer.





Inertial Measurement Unit

Classical Mechanics: Inertial Measurement Units

Absolute orientation from the accelerometer and magnetometer.

The IMUs measure *'linear'* accelerations and magnetic forces in the body's frame of reference:

$$\vec{a} = \left(a_i, a_j, a_k\right)^t \qquad \vec{m} = \left(m_i, m_j, m_k\right)^t$$

Choose the direction of gravity along the j-axis, and fix the local magnetic inclination α relative to the direction of the gravity vector:

$$\vec{g} = (0,1,0)^t \qquad \vec{b} = (\sin(\alpha), \cos(\alpha), 0)^t$$

Therefore the rotation matrix between the device co-ordinate frame and Earth's co-ordinate frame must satisfy both,

$$\vec{a} = \hat{R}\vec{g} \qquad \vec{m} = \hat{R}\vec{b}$$

Inertial Measurement Unit

Classical Mechanics: Example use case

Absolute orientation from the accelerometer and magnetometer.

The rotation matrix is partially determined

$$\hat{R} = \begin{pmatrix} \frac{m_i - a_i\cos(\alpha)}{\sin(\alpha)} & a_i & A \\ \frac{m_j - a_j\cos(\alpha)}{\sin(\alpha)} & a_j & B \\ \frac{m_k - a_k\cos(\alpha)}{\sin(\alpha)} & a_k & C \end{pmatrix}$$

The elements of the unknown vector A, B, C are given by the normalized cross product of the 2 known vectors.

The complete rotation matrix is

$$\hat{R} = \begin{pmatrix} \frac{m_i - a_i\cos(\alpha)}{\sin(\alpha)} & a_i & \pm\frac{a_k m_j - a_j m_k}{\sin(\alpha)} \\ \frac{m_j - a_j\cos(\alpha)}{\sin(\alpha)} & a_j & \pm\frac{a_i m_k - a_k m_i}{\sin(\alpha)} \\ \frac{m_k - a_k\cos(\alpha)}{\sin(\alpha)} & a_k & \pm\frac{a_j m_i - a_i m_j}{\sin(\alpha)} \end{pmatrix}$$

Euler Angles, Rodrigues Rotation Matrix and 6DoF

`Euler Angles, Rodrigues Rotation Matrix and 6DoF`

Euler Angles, Rodrigues Rotation Matrix and 6DoF

The roll-pitch-yaw angles:

- ψ_r roll angle:

$$\hat{R}(\psi_r) = \mathrm{Exp}\left[\psi_r \hat{\pi}_k\right]$$

- ψ_p pitch angle:

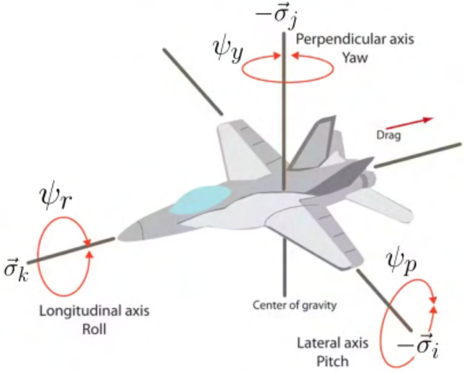
$$\hat{R}(\psi_p) = \mathrm{Exp}\left[\psi_p \hat{\pi}_i\right]$$

- ψ_y yaw angle:

$$\hat{R}(\psi_y) = \mathrm{Exp}\left[\psi_y \hat{\pi}_j\right]$$

define the rotation matrix

$$\hat{R} \equiv \hat{R}(\psi_y)\hat{R}(\psi_p)\hat{R}(\psi_r)$$



Euler Angles, Rodrigues Rotation Matrix and 6DoF

The rotation matrix is ordered from the right to the left, since we multiply the vector $\vec{r}$ from the right to the left.

$$\vec{r}' = [\text{yaw}*]\,[\text{pitch}*]\,[\text{roll}*]\,\vec{r}$$
$$\vec{r}' = \hat{R}(\psi_y)\hat{R}(\psi_p)\hat{R}(\psi_r)\vec{r}$$

Changing the multiplication order of the roll-pitch-yaw matrices, changes the final orientation.
→ non-commutative.

$$\hat{R}(\psi_r) = \begin{pmatrix} \cos(\psi_r) & -\sin(\psi_r) & 0 \\ \sin(\psi_r) & \cos(\psi_r) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\hat{R}(\psi_p) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi_p) & -\sin(\psi_p) \\ 0 & \sin(\psi_p) & \cos(\psi_p) \end{pmatrix}$$

$$\hat{R}(\psi_y) = \begin{pmatrix} \cos(\psi_y) & 0 & \sin(\psi_y) \\ 0 & 1 & 0 \\ -\sin(\psi_y) & 0 & \cos(\psi_y) \end{pmatrix}$$

Euler Angles, Rodrigues Rotation Matrix and 6DoF

Rodrigues' Rotation formula describes the rotation of a vector $\vec{v}$ around the unit-vector axis $\vec{k}$, through an angle θ.

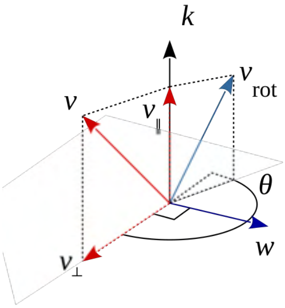
$$\vec{v}_{\text{rot}} = \vec{v}\cos(\theta) + \left(\vec{k}\times\vec{v}\right)\sin(\theta) + \vec{k}\left(\vec{k}\cdot\vec{v}\right)(1-\cos(\theta))$$

In matrix form the equivalent rotation is achieved with *Rodrigues' rotation matrix*

$$\hat{R} = \hat{\mathbb{1}} + \sin(\theta)\,\hat{k} + (1-\cos(\theta))\,\hat{k}^2$$

where $\hat{k}$ is the rotation axis vector expressed as a skew-symmetric matrix.

$$\vec{v}_{\text{rot}} = \hat{R}\vec{v}$$



Euler Angles, Rodrigues Rotation Matrix and 6DoF

Six degrees of freedom (6DOF), or sometimes six degrees of movement, refers to the six mechanical degrees of freedom of movement of a rigid body in three-dimensional space.

Rotation and Translation [Extrinsics]: A point cloud $\hat{X}$ in homogeneous coordinates is the matrix of the set of points $\vec{X}_i$

$$\hat{X} = \begin{pmatrix} \vec{X}_0 & \vec{X}_1 & \cdots & \vec{X}_i & \cdots & \end{pmatrix} \qquad \vec{X}_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix}$$

The extrinsics are described by the $3x4$ matrix, containing the SO(3) rotation matrix $\hat{R}$, and translation vector $\vec{t} = \left(t_x, t_y, t_z\right)^t$

$$\hat{X}' = \left[\hat{R} \,|\, \vec{t}\,\right] \hat{X}$$

Euler Angles, Rodrigues Rotation Matrix and 6DoF

$$\hat{X}' = \left[\hat{R} \mid \vec{t}\,\right] \hat{X}$$

$$\begin{pmatrix} X'_0 & X'_1 & \cdots & X'_i \\ Y'_0 & Y'_1 & \cdots & Y'_i \\ Z'_0 & Z'_1 & \cdots & Z'_i \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \end{pmatrix} \begin{pmatrix} X_0 & X_1 & \cdots & X_i \\ Y_0 & Y_1 & \cdots & Y_i \\ Z_0 & Z_1 & \cdots & Z_i \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

Problem Statement: Given the set of transformed co-ordinates $\hat{X}'$ and their initial positions $\hat{X}$, determine the extrinsics $\left[\hat{R} \mid \vec{t}\,\right]$ of the transformation.

Solution: The Kabsch algorithm, is a method for calculating the optimal rotation matrix, that minimizes the root mean squared deviation between two paired sets of points.

Euler Angles, Rodrigues Rotation Matrix and 6DoF

Problem Statement: Given $\hat{X}'$ and $\hat{X}$, determine the extrinsics.

Kabsch algorithm

1. Find the centroids of each set of points.
   $\rightarrow$ $\vec{c}' = \frac{1}{n}\sum_{i=0}^{n-1} \vec{X}'_i$ $\qquad \vec{c} = \frac{1}{n}\sum_{i=0}^{n-1} \vec{X}_i$
2. Transform each set of points relative to their centroids.
   $\rightarrow$ $\hat{P}' = \hat{X}' - \vec{c}'$ $\qquad \hat{P} = \hat{X} - \vec{c}$
3. Construct the covariance matrix $\hat{H} = \hat{P}'\hat{P}^t$
4. Calculate the Singular Valued Decomposition $\hat{H} = \hat{U}\hat{S}\hat{V}^t$
5. Calculate the Rotation Matrix $\hat{R} = \hat{V}\hat{U}^t$
6. Calculate the Translation $\vec{t} = \vec{c}' - \vec{c}$
7. Verify the solution: $\Delta\epsilon = \vec{X}' - \left(\left[\,\hat{R} \mid \vec{t}\,\right]\left(\hat{X} - \vec{c}\right) + \vec{c}\right)$

Euler Angles, Rodrigues Rotation Matrix and 6DoF

Problem Statement: Given the rotation matrix $\hat{R}$ determine the roll, pitch, and yaw angles (ψ_r, ψ_p, ψ_y).

```
In[27]:= MatrixExp[ψy πj].MatrixExp[ψp πi].MatrixExp[ψr πk] // MatrixForm
Out[27]//MatrixForm=
```

$$\begin{pmatrix} \cos[\psi r]\cos[\psi y]+\sin[\psi p]\sin[\psi r]\sin[\psi y] & -\cos[\psi y]\sin[\psi r]+\cos[\psi r]\sin[\psi p]\sin[\psi y] & \cos[\psi p]\sin[\psi y] \\ \cos[\psi p]\sin[\psi r] & \cos[\psi p]\cos[\psi r] & -\sin[\psi p] \\ \cos[\psi y]\sin[\psi p]\sin[\psi r]-\cos[\psi r]\sin[\psi y] & \cos[\psi r]\cos[\psi y]\sin[\psi p]+\sin[\psi r]\sin[\psi y] & \cos[\psi p]\cos[\psi y] \end{pmatrix}$$

$$\hat{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix}$$

$$\psi_p = \sin^{-1}(-R_{23})$$

$$\text{if} \quad \cos(\psi_p) > 10^{-6}: \qquad \psi_r = \tan^{-1}\left(\frac{R_{21}}{R_{22}}\right) \qquad \psi_y = \tan^{-1}\left(\frac{R_{13}}{R_{33}}\right)$$

$$\text{else:} \qquad \psi_r = 0 \qquad \psi_y = \tan^{-1}\left(\frac{R_{12}}{R_{32}}\right)$$

Euler Angles, Rodrigues Rotation Matrix and 6DoF

Problem Statement: Given the rotation matrix $\hat{R}$ determine the associated quaternion $\vec{Q}$.

$$\hat{R} = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{pmatrix}$$

Case #1: $\mathrm{Tr}\left[\hat{R}\right] > 0 \quad \rightarrow \quad \mathrm{Tr}\left[\hat{R}\right] = 3a^2 - b^2 - c^2 - d^2 = 4a^2 - 1$

$$\begin{aligned} a &= \tfrac{1}{2}\sqrt{\mathrm{Tr}\left[\hat{R}\right]+1} \\ b &= \tfrac{1}{4a}\left(R_{32} - R_{23}\right) \\ c &= \tfrac{1}{4a}\left(R_{13} - R_{31}\right) \\ d &= \tfrac{1}{4a}\left(R_{21} - R_{12}\right) \end{aligned}$$

Euler Angles, Rodrigues Rotation Matrix and 6DoF

Problem Statement: Given the rotation matrix $\hat{R}$ determine the associated quaternion $\vec{Q}$.

$$\hat{R} = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{pmatrix}$$

Case #2: R_{11} is max, $\rightarrow$ $R_{11} - R_{22} - R_{33} = 4b^2 - 1$

$$\begin{aligned} a &= \tfrac{1}{4b}\left(R_{32} - R_{23}\right) \\ b &= \tfrac{1}{2}\sqrt{R_{11} - R_{22} - R_{33} + 1} \\ c &= \tfrac{1}{4b}\left(R_{12} + R_{21}\right) \\ d &= \tfrac{1}{4b}\left(R_{13} + R_{31}\right) \end{aligned}$$

Cases #3 and #4 have similar constructions for R_{22}, and R_{33}, is max.

Euler Angles, Rodrigues Rotation Matrix and 6DoF

Problem Statement: Using GitHub co-pilot, generate code to implement all operations in this section.

1. Load and plot a `bunny.obj` file, using tkinter and matplotlib.
2. Assign an initial position that is different than the origin.
3. Rotate the `bunny.obj` around it's centroid using the Euler rotation matrix, and translate.
4. *Apply* a 2^{nd} global rotation, using Rodrigues' rotation matrix.
5. *Verify* 2^{nd} global rotation, using Rodrigues' rotation formula.
6. Make all operations tunable, and update the plot accordingly.
7. Track the total extrinsics using the Kabsch algorithm.
8. Calculate the Quaternion, Euler angles, and translation.
9. Verify all results.

The Hopf-Fibration

The Hopf-Fibration

The Hopf-Fibration

Consider the time dependent quaternion, as the product of a time dependent unit quaternion $\hat{U}(t)$, and a time independent initial state $\vec{Q}_0$, which is some randomly initialized unit quaternion.

$$\vec{\Psi}(t) = \vec{U}(t)\,\vec{Q}_0$$

For simplicity the unitary quaternion is defined

$$\vec{U}(t) = \begin{pmatrix} \cos\left(\frac{\beta(t)}{2}\right) \\ \sin\left(\frac{\beta(t)}{2}\right)\vec{r}_0 \end{pmatrix}$$

with $\vec{r}_0$ as a time independent unit vector describing a point on the 2-sphere $\mathbb{S}^2$. The time dependent angle $\beta(t) = 4\pi * t$.

The Hopf-Fibration

Consequently the time dependent unit quaternion

$$\vec{\Psi}(t) = \begin{pmatrix} a(t) \\ b(t) \\ c(t) \\ d(t) \end{pmatrix}$$

and $a^2 + b^2 + c^2 + d^2 = 1$, for all t.
The time dependent quaternion $\vec{\Psi}$ traces a path on the hypersphere $\mathbb{S}^3$ embedded in $\mathbb{R}^4$.

$$\vec{\Psi} \in \mathbb{S}^3 \subset \mathbb{R}^4, \mathbb{C}^2$$

The Hopf-Fibration

The corresponding SO(3) operator is as before

$$\hat{\Psi} = \begin{pmatrix} a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2 \end{pmatrix}$$

The rows, and columns of $\hat{\Psi}$ describe a set of 6 points on the 2-sphere $\mathbb{S}^2$. Labelling the rows and columns of $\hat{\Psi}$ respectively as

$$\hat{\Psi} = \begin{pmatrix} \vec{\mathscr{I}}_l \\ \vec{\mathscr{J}}_l \\ \vec{\mathscr{K}}_l \end{pmatrix} \qquad \hat{\Psi} = \begin{pmatrix} \vec{\mathscr{I}}_r & \vec{\mathscr{J}}_r & \vec{\mathscr{K}}_r \end{pmatrix}$$

The vectors $\vec{X}_\bullet = (X^i_\bullet, X^j_\bullet, X^k_\bullet)$, for $\bullet = l, r$, and $X = (\mathscr{I}, \mathscr{J}, \mathscr{K})$.

The Hopf-Fibration

We may access the individual vectors according to

$$
\begin{aligned}
\hat{\mathscr{I}}_l &= \hat{\Psi}^t \, \hat{\pi}_i \, \hat{\Psi} & \qquad \hat{\mathscr{I}}_r &= \hat{\Psi} \, \hat{\pi}_i \, \hat{\Psi}^t \\
\hat{\mathscr{J}}_l &= \hat{\Psi}^t \, \hat{\pi}_j \, \hat{\Psi} & \qquad \hat{\mathscr{J}}_r &= \hat{\Psi} \, \hat{\pi}_j \, \hat{\Psi}^t \\
\hat{\mathscr{K}}_l &= \hat{\Psi}^t \, \hat{\pi}_k \, \hat{\Psi} & \qquad \hat{\mathscr{K}}_r &= \hat{\Psi} \, \hat{\pi}_k \, \hat{\Psi}^t
\end{aligned}
$$

Via these 6 projections, the quaternion expressed as the rotation matrix:

$$\hat{\Psi} \in \mathbb{S}^2 \subset \mathbb{R}^3$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_i$ axis, in the left Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{l}_i\right]\vec{\Psi} = \begin{pmatrix} \cos\left(\frac{\vec{\varphi}}{2}\right) & -\sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 \\ \sin\left(\frac{\vec{\varphi}}{2}\right) & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 \\ 0 & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & -\sin\left(\frac{\vec{\varphi}}{2}\right) \\ 0 & 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) & \cos\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_j$ axis, in the left Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{l}_j\right]\vec{\Psi} = \begin{pmatrix} \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & -\sin\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & -\sin\left(\frac{\vec{\varphi}}{2}\right) \\ \sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$(a', b', c', d') \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'} \right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_k$ axis, in the left Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{l}_k\right]\vec{\Psi} = \begin{pmatrix} \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 & -\sin\left(\frac{\vec{\varphi}}{2}\right) \\ 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & -\sin\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ \sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \mapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \mapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_i$ axis, in the right Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{r}_i\right]\vec{\Psi} = \begin{pmatrix} \cos\left(\frac{\vec{\varphi}}{2}\right) & \sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 \\ -\sin\left(\frac{\vec{\varphi}}{2}\right) & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 \\ 0 & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & -\sin\left(\frac{\vec{\varphi}}{2}\right) \\ 0 & 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) & \cos\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_j$ axis, in the right Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{r}_j\right]\vec{\Psi} = \begin{pmatrix} \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & -\sin\left(\frac{\vec{\varphi}}{2}\right) \\ -\sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \mapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \mapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_k$ axis, in the right Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{r}_k\right]\vec{\Psi} = \begin{pmatrix} \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) \\ 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) & -\sin\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ 0 & \sin\left(\frac{\vec{\varphi}}{2}\right) & \cos\left(\frac{\vec{\varphi}}{2}\right) & 0 \\ -\sin\left(\frac{\vec{\varphi}}{2}\right) & 0 & 0 & \cos\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \mapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \mapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_i$ axis, in the left Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{l}_i\right]\vec{\Psi} = \begin{pmatrix} a\cos\left(\frac{\vec{\varphi}}{2}\right) - b\sin\left(\frac{\vec{\varphi}}{2}\right) \\ b\cos\left(\frac{\vec{\varphi}}{2}\right) + a\sin\left(\frac{\vec{\varphi}}{2}\right) \\ c\cos\left(\frac{\vec{\varphi}}{2}\right) - d\sin\left(\frac{\vec{\varphi}}{2}\right) \\ d\cos\left(\frac{\vec{\varphi}}{2}\right) + c\sin\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \mapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \mapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_j$ axis, in the left Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{l}_j\right]\vec{\Psi} = \begin{pmatrix} a\cos\left(\frac{\vec{\varphi}}{2}\right) - c\sin\left(\frac{\vec{\varphi}}{2}\right) \\ b\cos\left(\frac{\vec{\varphi}}{2}\right) + d\sin\left(\frac{\vec{\varphi}}{2}\right) \\ c\cos\left(\frac{\vec{\varphi}}{2}\right) + a\sin\left(\frac{\vec{\varphi}}{2}\right) \\ d\cos\left(\frac{\vec{\varphi}}{2}\right) - b\sin\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_k$ axis, in the left Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{l}_k\right]\vec{\Psi} = \begin{pmatrix} a \cos\left(\frac{\vec{\varphi}}{2}\right) - d \sin\left(\frac{\vec{\varphi}}{2}\right) \\ b \cos\left(\frac{\vec{\varphi}}{2}\right) - c \sin\left(\frac{\vec{\varphi}}{2}\right) \\ c \cos\left(\frac{\vec{\varphi}}{2}\right) + b \sin\left(\frac{\vec{\varphi}}{2}\right) \\ d \cos\left(\frac{\vec{\varphi}}{2}\right) + a \sin\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_i$ axis, in the right Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{r}_i\right]\vec{\Psi} = \begin{pmatrix} a\cos\left(\frac{\vec{\varphi}}{2}\right) + b\sin\left(\frac{\vec{\varphi}}{2}\right) \\ b\cos\left(\frac{\vec{\varphi}}{2}\right) - a\sin\left(\frac{\vec{\varphi}}{2}\right) \\ c\cos\left(\frac{\vec{\varphi}}{2}\right) - d\sin\left(\frac{\vec{\varphi}}{2}\right) \\ d\cos\left(\frac{\vec{\varphi}}{2}\right) + c\sin\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_j$ axis, in the right Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{r}_j\right]\vec{\Psi} = \begin{pmatrix} a\cos\left(\frac{\vec{\varphi}}{2}\right) + c\sin\left(\frac{\vec{\varphi}}{2}\right) \\ b\cos\left(\frac{\vec{\varphi}}{2}\right) + d\sin\left(\frac{\vec{\varphi}}{2}\right) \\ c\cos\left(\frac{\vec{\varphi}}{2}\right) - a\sin\left(\frac{\vec{\varphi}}{2}\right) \\ d\cos\left(\frac{\vec{\varphi}}{2}\right) - b\sin\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_k$ axis, in the right Cayley basis

$$\vec{\Psi}' = \exp\left[\frac{\vec{\varphi}}{2}\hat{r}_k\right]\vec{\Psi} = \begin{pmatrix} a \cos\left(\frac{\vec{\varphi}}{2}\right) + d \sin\left(\frac{\vec{\varphi}}{2}\right) \\ b \cos\left(\frac{\vec{\varphi}}{2}\right) - c \sin\left(\frac{\vec{\varphi}}{2}\right) \\ c \cos\left(\frac{\vec{\varphi}}{2}\right) + b \sin\left(\frac{\vec{\varphi}}{2}\right) \\ d \cos\left(\frac{\vec{\varphi}}{2}\right) - a \sin\left(\frac{\vec{\varphi}}{2}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \mapsto \mathbb{R}^3$$

defined by

$$(a', b', c', d') \mapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_i$ axis, in the left Cayley basis

$$\hat{\Psi}' = \begin{pmatrix} \mathscr{I}_l^i & \mathscr{I}_l^j & \mathscr{I}_l^k \\ \mathscr{J}_l^i \mathrm{c}(\vec{\varphi}) - \mathscr{K}_l^i \mathrm{s}(\vec{\varphi}) & \mathscr{J}_l^j \mathrm{c}(\vec{\varphi}) - \mathscr{K}_l^j \mathrm{s}(\vec{\varphi}) & \mathscr{J}_l^k \mathrm{c}(\vec{\varphi}) - \mathscr{K}_l^k \mathrm{s}(\vec{\varphi}) \\ \mathscr{K}_l^i \mathrm{c}(\vec{\varphi}) + \mathscr{J}_l^i \mathrm{s}(\vec{\varphi}) & \mathscr{K}_l^j \mathrm{c}(\vec{\varphi}) + \mathscr{J}_l^j \mathrm{s}(\vec{\varphi}) & \mathscr{K}_l^k \mathrm{c}(\vec{\varphi}) + \mathscr{J}_l^k \mathrm{s}(\vec{\varphi}) \end{pmatrix}$$

with $(\mathrm{c}(\vec{\varphi}), \mathrm{s}(\vec{\varphi})) = (\cos(\vec{\varphi}), \sin(\vec{\varphi}))$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_j$ axis, in the left Cayley basis

$$\hat{\Psi}' = \begin{pmatrix} \mathscr{I}_l^i \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{K}_l^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{I}_l^j \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{K}_l^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{I}_l^k \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{K}_l^k \mathrm{s}\left(\vec{\varphi}\right) \\ \mathscr{J}_l^i & \mathscr{J}_l^j & \mathscr{J}_l^k \\ \mathscr{K}_l^i \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{I}_l^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_l^j \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{I}_l^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_l^k \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{I}_l^k \mathrm{s}\left(\vec{\varphi}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_k$ axis, in the left Cayley basis

$$\hat{\Psi}' = \begin{pmatrix} \mathscr{I}_l^i \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{J}_l^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{I}_l^j \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{J}_l^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{I}_l^k \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{J}_l^k \mathrm{s}\left(\vec{\varphi}\right) \\ \mathscr{J}_l^i \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{I}_l^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_l^j \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{I}_l^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_l^k \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{I}_l^k \mathrm{s}\left(\vec{\varphi}\right) \\ \mathscr{K}_l^i & \mathscr{K}_l^j & \mathscr{K}_l^k \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \mapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \mapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_i$ axis, in the right Cayley basis

$$\hat{\Psi}' = \begin{pmatrix} \mathscr{I}_r^i & \mathscr{J}_r^i \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{K}_r^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_r^i \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{J}_r^i \mathrm{s}\left(\vec{\varphi}\right) \\ \mathscr{I}_r^j & \mathscr{J}_r^j \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{K}_r^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_r^j \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{J}_r^j \mathrm{s}\left(\vec{\varphi}\right) \\ \mathscr{I}_r^k & \mathscr{J}_r^k \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{K}_r^k \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_r^k \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{J}_r^k \mathrm{s}\left(\vec{\varphi}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \mapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \mapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_j$ axis, in the right Cayley basis

$$\hat{\Psi}' = \begin{pmatrix} \mathscr{I}_r^i \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{K}_r^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_r^i & \mathscr{K}_r^i \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{I}_r^i \mathrm{s}\left(\vec{\varphi}\right) \\ \mathscr{I}_r^j \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{K}_r^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_r^j & \mathscr{K}_r^j \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{I}_r^j \mathrm{s}\left(\vec{\varphi}\right) \\ \mathscr{I}_r^k \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{K}_r^k \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_r^k & \mathscr{K}_r^k \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{I}_r^k \mathrm{s}\left(\vec{\varphi}\right) \end{pmatrix}$$

The Hopf-Fibration

The Stereographic projection of the quaternion is the map

$$\mathbb{S}^3 / (1,0,0,0) \longmapsto \mathbb{R}^3$$

defined by

$$\left(a', b', c', d'\right) \longmapsto \left(\frac{b'}{1-a'}, \frac{c'}{1-a'}, \frac{d'}{1-a'}\right)$$

Rotate the quaternion $\vec{\Psi}$, through all angles $\vec{\varphi}$, for $\vec{\varphi} \in [0, 4\pi]$, about the $\vec{\sigma}_k$ axis, in the right Cayley basis

$$\hat{\Psi}' = \begin{pmatrix} \mathscr{I}_r^i \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{J}_r^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_r^i \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{I}_r^i \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_r^i \\ \mathscr{I}_r^j \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{J}_r^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_r^j \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{I}_r^j \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_r^j \\ \mathscr{I}_r^k \mathrm{c}\left(\vec{\varphi}\right) - \mathscr{J}_r^k \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{J}_r^k \mathrm{c}\left(\vec{\varphi}\right) + \mathscr{I}_r^k \mathrm{s}\left(\vec{\varphi}\right) & \mathscr{K}_r^k \end{pmatrix}$$

The Hopf-Fibration

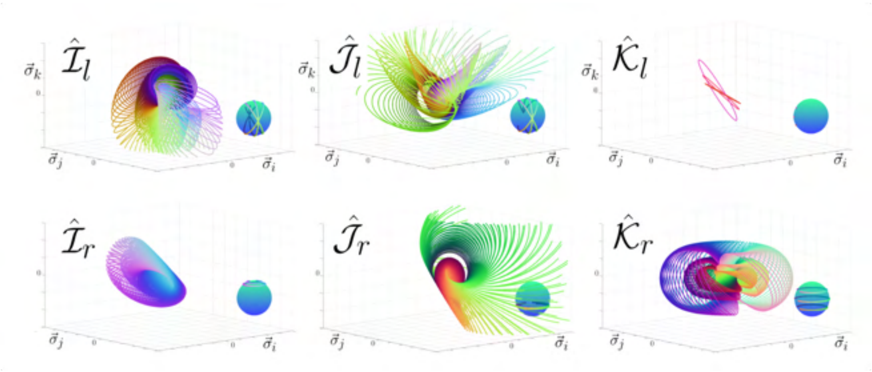
At any point in time t we can generate a total of 6 great circles for the quaternion $\vec{\Psi}$. There is one great circle $\mathbb{S}^1$ for each axis. The great circles of the Hopf mapping

$$\mathbb{S}^3 \xrightarrow{\mathbb{S}^1} \mathbb{S}^2$$

can be visualized using the stereographic projection of the quaternion.

The Hopf-Fibration

$$\mathbb{S}^3 \xrightarrow{\mathbb{S}^1} \mathbb{S}^2$$



This figure is generated for set of quaternions (defined in a manner different to the preceding section, but is useful for a visual representation of the hopf fibration). For more information see arXiv:1601.02569

The Hopf-Fibration

The time dependent case.

In the SO(3) picture, the quaternion evolves as

$$\dot{\hat{\Psi}} = \hat{\mathscr{H}}^{+}\hat{\Psi} \qquad \dot{\hat{\Psi}}^{t} = \hat{\mathscr{H}}^{-}\hat{\Psi}^{t}$$

$$\dot{\vec{R}} = \hat{\mathscr{H}}^{+}\vec{R} \qquad \dot{\vec{L}} = \hat{\mathscr{H}}^{-}\vec{L}$$

where $\vec{R},\vec{L}$ represent the column vectors of the quaternion, following the right and left Cayley rotations respectively.

Denoting $\vec{X} = \vec{R},\vec{L}$, and $\vec{\mathscr{H}}$ is the corresponding Hamiltonian. The derivatives of the dynamic and geometric phases are defined:

$$\dot{\xi} = \vec{X}\cdot\vec{\mathscr{H}} \qquad \dot{\gamma} = \frac{\dot{\vec{X}}\cdot\dot{\vec{\mathscr{H}}} - \left(\dot{\vec{X}}\cdot\dot{\vec{X}}\right)\left(\vec{X}\cdot\vec{\mathscr{H}}\right)}{\vec{\mathscr{H}}\cdot\vec{\mathscr{H}} - \left(\vec{X}\cdot\vec{\mathscr{H}}\right)^{2}}$$

The Hopf-Fibration

The dynamic and geometric phases are defined

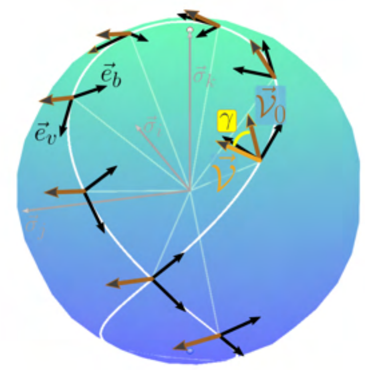
$$\xi(t) \equiv \int_0^t dt' \, \dot{\xi}(t') \qquad \gamma(t) \equiv \int_0^t dt' \, \dot{\gamma}(t')$$

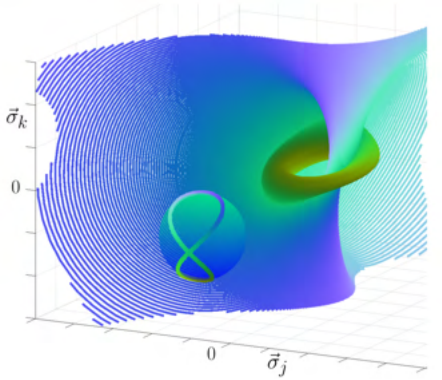
The global phase is the sum of the dynamic and geometric phases

$$\omega(t) = \xi(t) + \gamma(t)$$

Closed paths on $\mathbb{S}^2$ are generated from $t = 0$ until $t = 4\pi$.
For all closed paths the global phase is an integer multiple of 2π.

The Hopf-Fibration





Example of a closed path on the 2-sphere, showing the geometric phase in the Darboux tangent frame (left). The $\mathbb{S}^2$ path has an associated great circle $\mathbb{S}^1$ for all times t (right).

The Hopf-Fibration

Further study.

- Stokes theorem, the Gauss-Bonnet theorem.
- @keenancrane has an excellent resource of lectures available which are most suited to studying the topology of the quaternion and the hopf-fibration.

The Hopf-Fibration

The landscape of the quaternion.

According to to the Adam's theorem, the extensions of the hopf-fibration are limited to

$$\mathbb{S}^3 \xrightarrow{\mathbb{S}^1} \mathbb{S}^2 \qquad \mathbb{S}^7 \xrightarrow{\mathbb{S}^3} \mathbb{S}^4 \qquad \mathbb{S}^{15} \xrightarrow{\mathbb{S}^7} \mathbb{S}^8$$

where

- $\mathbb{S}^1$ the complex numbers
- $\mathbb{S}^3$ the quaternions $\rightarrow$ non-commutative.
- $\mathbb{S}^7$ the octonians $\rightarrow$ non-commutative and non-associative.
- $\mathbb{S}^{15}$ the sedenians $\rightarrow$ non-commutative, non-associative, non-alternative.

J. F. Adams "On the non-existence of elements of Hopf invariant one" The Annals of Mathematics, 72(1):20(**104**) (1960). J. F. Adams, M. F. Atiyah "K-Theory and the Hopf Invariant" The Quartely Journal of Mathematics, 17(1):31(**38**) (1966).

The Hopf-Fibration

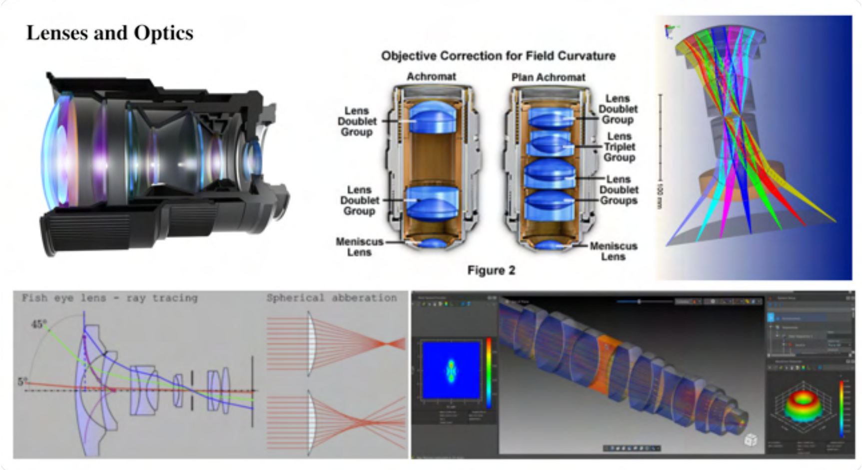
The 4^{th} dimension.

CarlSagan explains the 4^{th} dimension using the example of how the 3^{rd} dimension would appear to a 2 dimensional being. [youtube link]

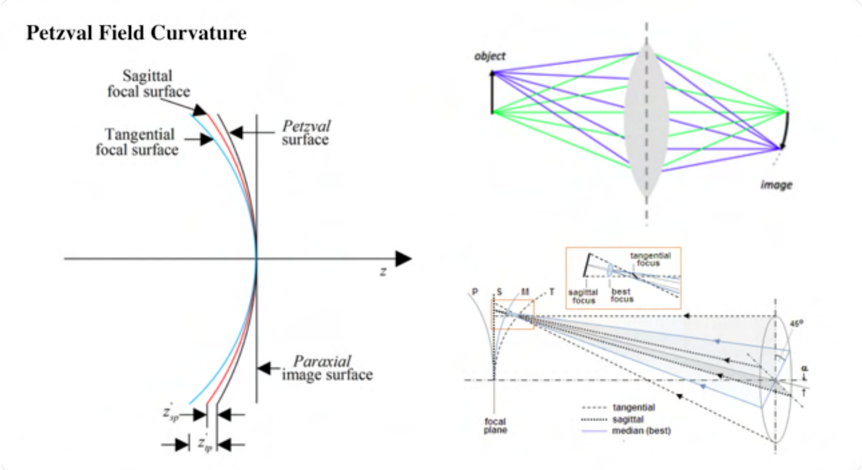
Section Outline

`Point Centric Camera Model`

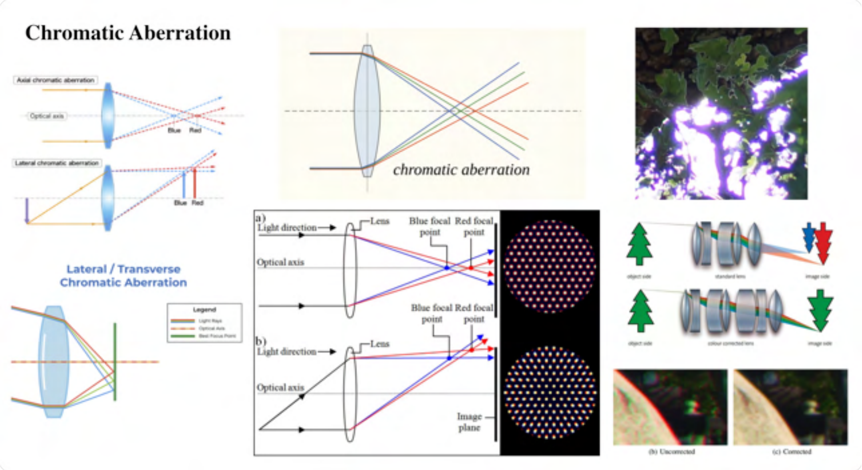
Section Outline



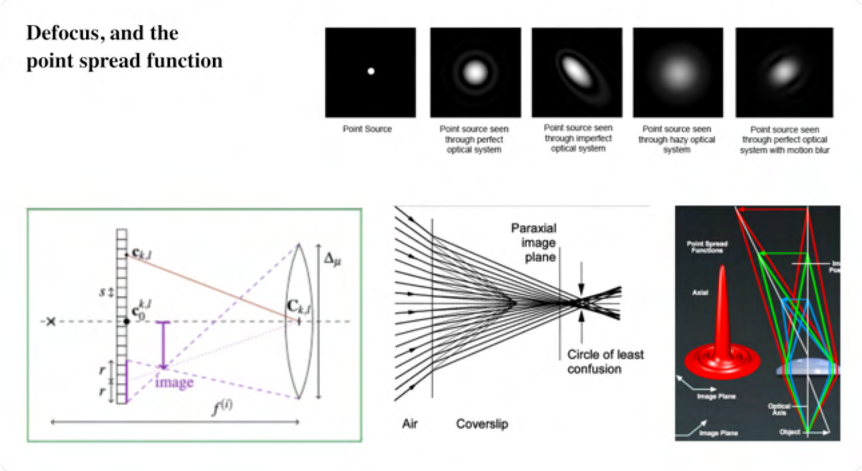
Section Outline



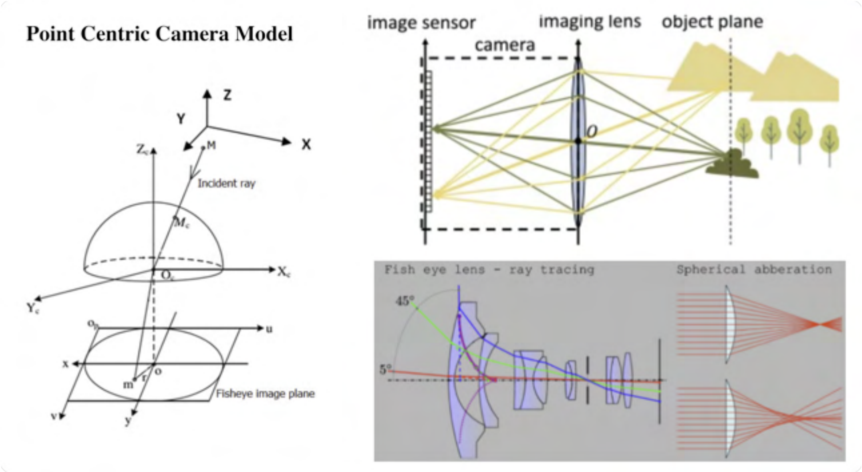
Section Outline



Section Outline



Section Outline



Section Outline

$$\hat{X} \xmapsto{\text{extrinsics}} \hat{P} \xmapsto{\text{lens distortion}} \hat{p} \xmapsto{\text{sensor tilt}} \hat{p}' \xmapsto{\text{intrinsics}} \hat{x}$$

Stages of the camera model

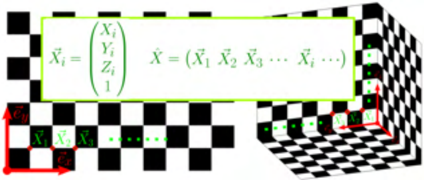
1. $\hat{X}$ → A point cloud of known coordinates, centred on/near the world origin.
2. $\hat{P}$ → The point cloud is transformed to world coordinates according to the 6 degrees of freedom.
3. $\hat{p}$ → Points in the 3D world are transformed to 2D lens distorted coordinates, per the fish-eye lens.
4. $\hat{p}'$ → Misalignment's between the lens' optical axis and sensor plane normal, distorts the image to sensor tilted coordinates.
5. $\hat{x}$ → Points in camera coordinates are transformed to pixel coordinates, via the camera's intrinsic matrix $\hat{K}$.

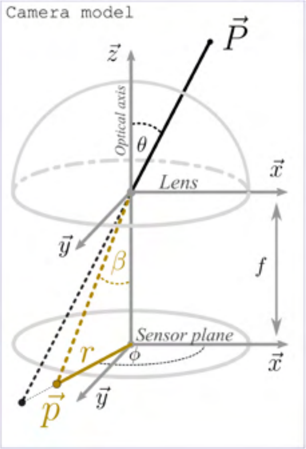
World Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

World co-ordinates:

- $\hat{X}$ are the checkerboard corner co-ordinates in millimetres.
- The checkerboard is initially located in the $z = 0$ plane, with the board center at $(x, y) = (0, 0)$.



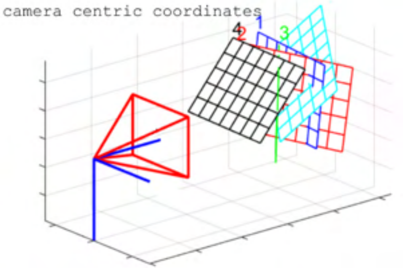


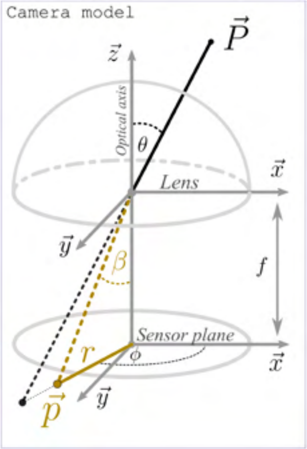
World Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

World co-ordinates:

- $\hat{P}$ are the world co-ordinates after rotation and translation into the camera's field of view.
- $\hat{P} = [\, \hat{R} \mid \vec{t} \,] \, \hat{X}$



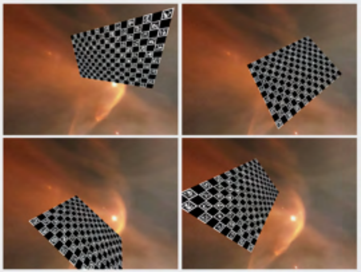


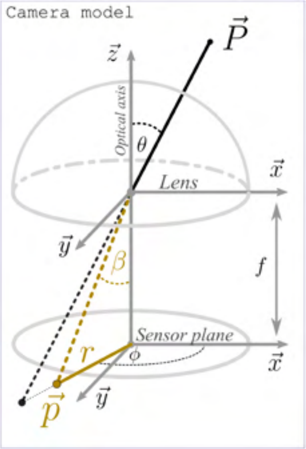
World Coordinates

$$\color{blue}{\hat{X} \xrightarrow{\text{extrinsics}} \hat{P}} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

World co-ordinates:

- $\hat{P}$ are the world co-ordinates after rotation and translation into the camera's field of view.
- $\hat{P} = [\, \hat{R} \,|\, \vec{t}\,]\, \hat{X}$



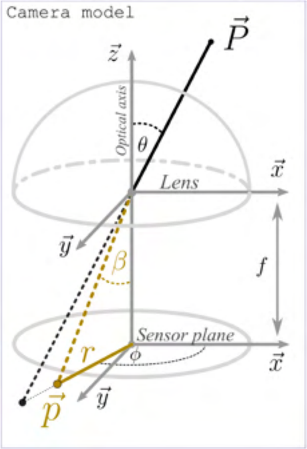


Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Lens Distortion:

- The camera lens compresses a large field of view.
- Lens distortion is a function of
  (1) field angle relative to the optical axis
  (2) the focal length of the camera
- There are 4 fundamental fish eye lens models.
- Commercial lenses typically deviate from the standard models

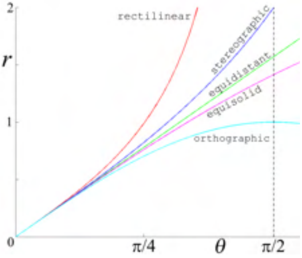


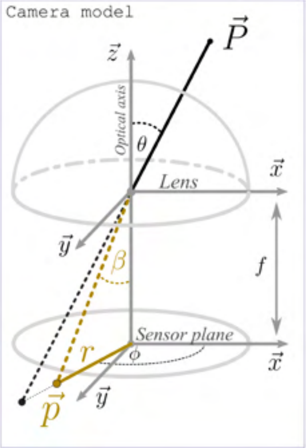
Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Fish-Eye Lens Models:

- Stereographic: $r(\theta) = 2f\tan\left(\frac{\theta}{2}\right)$
- Equidistant: $r(\theta) = f\theta$
- Equisolid: $r(\theta) = 2f\sin\left(\frac{\theta}{2}\right)$
- Orthographic: $r(\theta) = f\sin(\theta)$



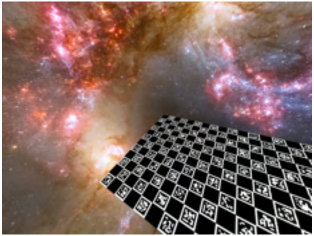


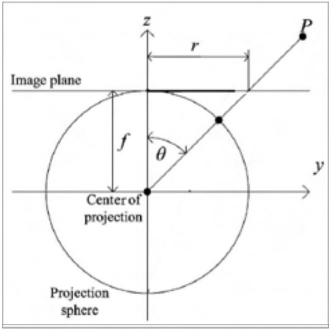
Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Fish-Eye Lens Models:

- Stereographic: $r(\theta) = 2f\tan\left(\frac{\theta}{2}\right)$
- Equidistant: $r(\theta) = f\theta$
- Equisolid: $r(\theta) = 2f\sin\left(\frac{\theta}{2}\right)$
- Orthographic: $r(\theta) = f\sin(\theta)$





Rectilinear Lens Model

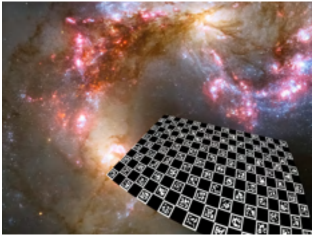
Hughes *et al.* "Accuracy of fish-eye lens models" Applied Optics **49**(17) *3338-3347*.

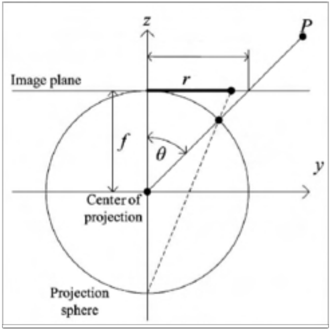
Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Fish-Eye Lens Models:

- Stereographic: $r(\theta) = 2f\tan\left(\frac{\theta}{2}\right)$
- Equidistant: $r(\theta) = f\theta$
- Equisolid: $r(\theta) = 2f\sin\left(\frac{\theta}{2}\right)$
- Orthographic: $r(\theta) = f\sin(\theta)$





Stereographic Lens Model

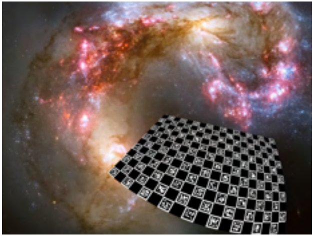
Hughes *et al.* "Accuracy of fish-eye lens models" Applied Optics **49**(17) *3338-3347*.

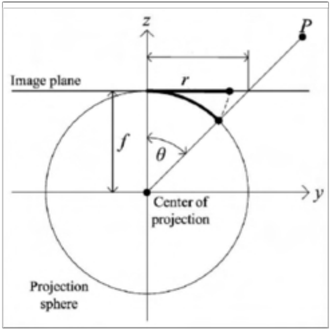
Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Fish-Eye Lens Models:

- Stereographic: $r(\theta) = 2f\tan\left(\frac{\theta}{2}\right)$
- Equidistant: $r(\theta) = f\theta$
- Equisolid: $r(\theta) = 2f\sin\left(\frac{\theta}{2}\right)$
- Orthographic: $r(\theta) = f\sin(\theta)$





Equidistant Lens Model

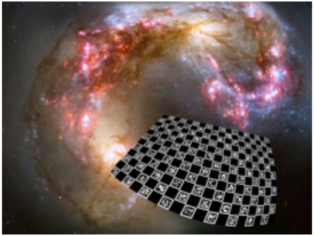
Hughes *et al.* "Accuracy of fish-eye lens models" Applied Optics **49**(17) *3338-3347*.

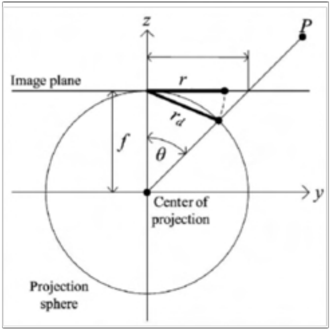
Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Fish-Eye Lens Models:

- Stereographic: $r(\theta) = 2f \tan\left(\frac{\theta}{2}\right)$
- Equidistant: $r(\theta) = f\theta$
- Equisolid: $r(\theta) = 2f \sin\left(\frac{\theta}{2}\right)$
- Orthographic: $r(\theta) = f \sin(\theta)$





Equisolid Lens Model

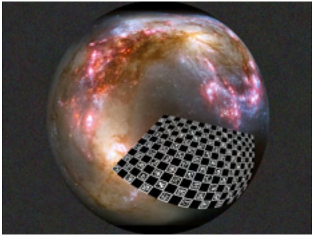
Hughes *et al.* "Accuracy of fish-eye lens models" Applied Optics **49**(17) *3338-3347*.

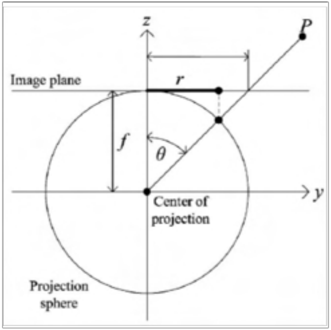
Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Fish-Eye Lens Models:

- Stereographic: $r(\theta) = 2f \tan\left(\frac{\theta}{2}\right)$
- Equidistant: $r(\theta) = f\theta$
- Equisolid: $r(\theta) = 2f \sin\left(\frac{\theta}{2}\right)$
- Orthographic: $r(\theta) = f \sin(\theta)$





Orthographic Lens Model

Hughes *et al.* "Accuracy of fish-eye lens models" Applied Optics **49**(17) *3338-3347*.

Lens Distorted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

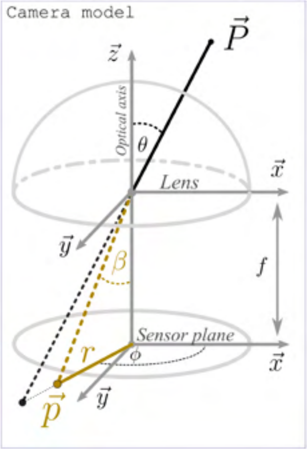
Lens distorted co-ordinates:

$$\theta = \tan^{-1}\left(\frac{\sqrt{P_x^2 + P_y^2}}{P_z}\right)$$

$$\phi = \tan^{-1}\left(\frac{P_y}{P_x}\right)$$

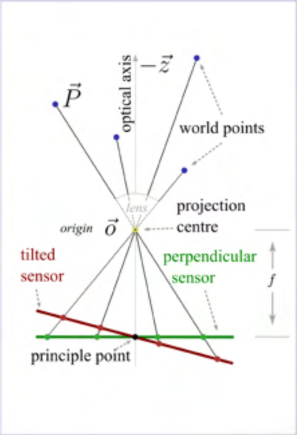
$$\vec{p} = \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} r(\theta)\cos(\phi) \\ r(\theta)\sin(\phi) \\ 1 \end{pmatrix}$$

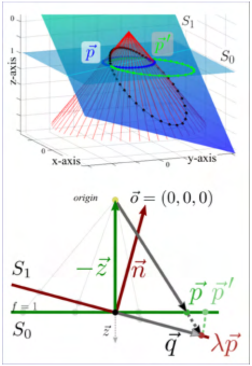
$$\hat{p} = \begin{pmatrix} \vec{p}_1 & \vec{p}_2 & \cdots & \vec{p}_i & \cdots \end{pmatrix}$$



Sensor Tilted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$





O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Sensor Tilted Coordinates

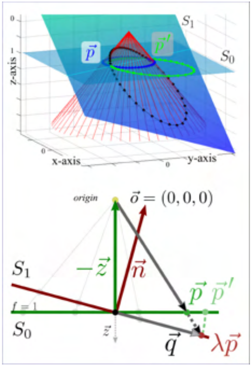
$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

The sensor plane has 2 positions

- S_0 perpendicular sensor
- S_1 tilted sensor

Photons emanting from the world points $\vec{P}$ are projected through the lens elements, to impact the perpendicular sensor S_0 at the points $\vec{p}$.

$$\vec{z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad \vec{n} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \qquad \vec{p} = \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$



O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Sensor Tilted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

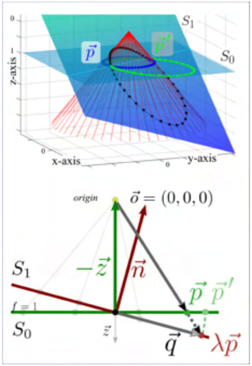
The projected point forms a vector $\lambda\vec{p}$ touching the tilted plane S_1, connecting with $\vec{q}$ in the local co-ordinates of S_1.

$$\vec{q} = \lambda\vec{p} - \vec{z}$$

$$\vec{n}\cdot\vec{q} = 0 = \lambda\vec{n}\cdot\vec{p} - \vec{n}\cdot\vec{z}$$

$$\lambda = \frac{\vec{n}\cdot\vec{z}}{\vec{n}\cdot\vec{p}}$$

$$\lambda = \frac{n_z}{n_x p_x + n_y p_y + n_z}$$



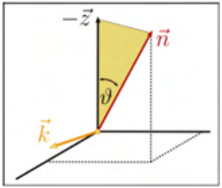
O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Sensor Tilted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

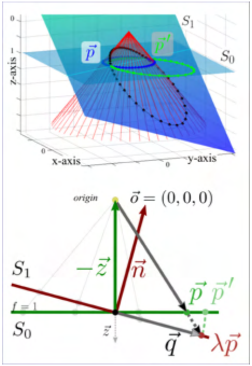
The rotation angle, and rotation axis:

$$\vartheta = \cos^{-1}(\vec{n}\cdot(-\vec{z}))$$
$$\vec{n}\times\vec{z} = |\vec{n}||\vec{z}|\sin(\vartheta)$$



The unit rotation axis is:

$$\vec{k} = \frac{1}{\sin(\vartheta)}\begin{pmatrix}-n_y\\ n_x\\ 0\end{pmatrix} = \frac{\vec{k}'}{\sin(\vartheta)}$$



O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Sensor Tilted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

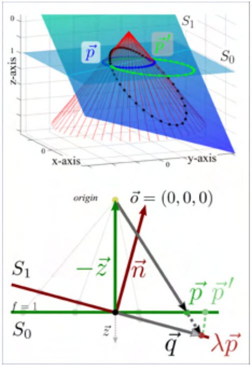
Rodrigues rotation matrix:

$$\hat{R} = \hat{\mathbb{1}} + \sin(\vartheta)\,\hat{k} + (1 - \cos(\vartheta))\,\hat{k}^2$$

Let $\hat{k} = \frac{1}{\sin(\vartheta)}\hat{k}'$, and $n_z = -\cos(\vartheta)$.

$$\hat{R} = \hat{\mathbb{1}} + \hat{k}' + \frac{1 + n_z}{(1 + n_z)(1 - n_z)}\hat{k}'^2$$

$$\hat{R} = \begin{pmatrix} 1 + \frac{n_x^2}{n_z - 1} & \frac{n_x n_y}{n_z - 1} & n_x \\ \frac{n_x n_y}{n_z - 1} & 1 + \frac{n_y^2}{n_z - 1} & n_y \\ -n_x & -n_y & -n_z \end{pmatrix}$$



O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Sensor Tilted Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

To perform a rotation around the projective plane origin:

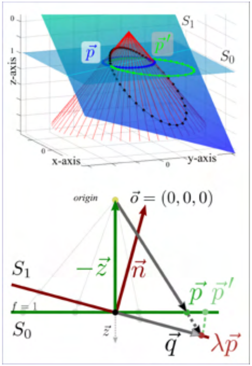
shift to origin → rotate → shift back

$$\vec{p}\,' = \hat{R}\left(\lambda\vec{p} - \vec{z}\right) + \vec{z}$$

The tilt shifted map $\vec{p} \mapsto \vec{p}\,'$:

$$p_x{}' = \frac{(n_x^2 + n_z(n_z - 1))p_x + n_x n_y p_y}{\left(n_x p_x + n_y p_y + n_z\right)(n_z - 1)}$$

$$p_y{}' = \frac{(n_y^2 + n_z(n_z - 1))p_y + n_x n_y p_x}{\left(n_x p_x + n_y p_y + n_z\right)(n_z - 1)}$$



O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Pixel Coordinates

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Transform to pixel coordinates via the camera matrix.

$$\hat{x} = \hat{K}\hat{p}'$$

$$\hat{K} = \begin{pmatrix} f & 0 & o_x \\ 0 & \alpha f & o_y \\ 0 & 0 & 1 \end{pmatrix}$$

f is the focal length of the camera in pixels, ($\alpha \approx 1$).
The optical axis is $(o_x, o_y) = (x_c + dX, y_c + dY)$, where (x_c, y_c) is the image center

$$(x_c, y_c) = \left(\frac{w}{2} - \frac{1}{2}, \frac{h}{2} - \frac{1}{2}\right)$$

$(dX, dY) \rightarrow$ offset between the len's optical axis and image center.

Pixel Coordinates

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

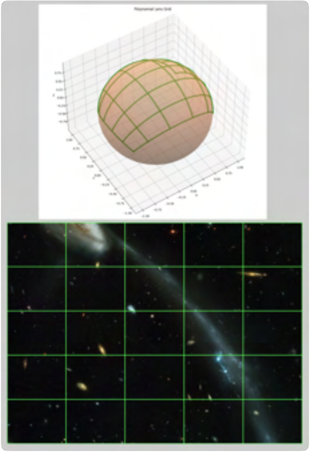
The Inverse Camera Model.

- Each Pixel is a vector in 3D space
- Pixel’s FOV is max at image center

The pixel vector intersects an object in 3D space, and determines the colour.

Inverse camera model is required for:

- Ray Tracing Virtual Camera
- Optimization & Camera Calibration



Pixel Coordinates

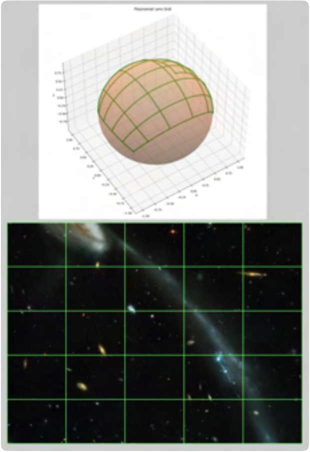
$$\hat{x} \xmapsto{\text{intrinsics}} \hat{p}' \xmapsto{\text{sensor untilt}} \hat{p} \xmapsto{\text{lens undistortion}} \hat{P} \xmapsto{\text{extrinsics}} \hat{X}$$

The Inverse Camera Model.

- Sensor of (height, width) = (h, w).
- Create an array of $n = h * w$ pixel coordinates.

$$\hat{x} = \begin{pmatrix} x_0 & x_1 & \cdots & x_i & \cdots & x_{n-1} \\ y_0 & y_1 & \cdots & y_i & \cdots & y_{n-1} \\ 1 & 1 & \cdots & 1 & \cdots & 1 \end{pmatrix}$$

- $(x_0, y_0) = (0, 0)$
- $(x_{n-1}, y_{n-1}) = (w - 1, h - 1)$
- Center: $(x_c, y_c) = (\frac{w}{2} - \frac{1}{2}, \frac{h}{2} - \frac{1}{2})$



Pixel Coordinates

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

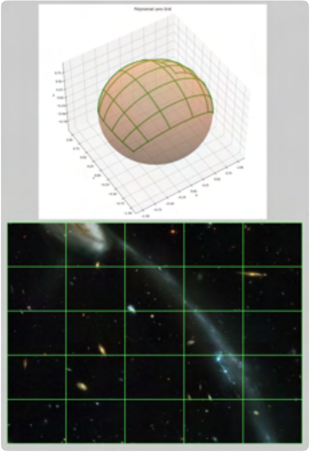
The Inverse Camera Model.

- Center: $(x_c, y_c) = (\frac{w}{2} - \frac{1}{2}, \frac{h}{2} - \frac{1}{2})$
- The inverse camera matrix:

$$\hat{K}^{-1} = \begin{pmatrix} \frac{1}{f} & 0 & -\frac{o_x}{f} \\ 0 & \frac{1}{\alpha f} & -\frac{o_y}{\alpha f} \\ 0 & 0 & 1 \end{pmatrix}$$

- Transform to camera co-ordinates:

$$\hat{p}' = \hat{K}^{-1}\,\hat{x}$$



Sensor Untilt

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

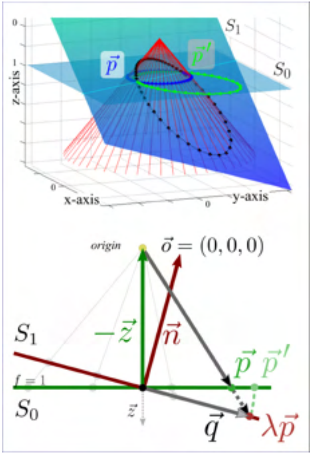
Remove Sensor Tilt:

shift to origin $\rightarrow$ rotation^{-1} $\rightarrow$ shift back

$$\vec{p} = \lambda' \left(\hat{R}^t \left(\vec{p}' - \vec{z}\right) + \vec{z}\right)$$

The scale factor is the pinhole projection

$$\lambda' = \left(n_x p'_x + n_y p'_y + 1\right)^{-1}$$



O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Sensor Untilt

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

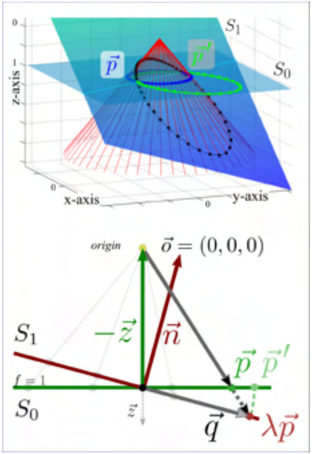
Remove Sensor Tilt:

shift to origin $\rightarrow$ rotation^{-1} $\rightarrow$ shift back

The tilt corrected map $\vec{p}' \mapsto \vec{p}$:

$$p_x = \frac{(n_x^2 + n_z - 1)\,p'_x + n_x n_y p'_y}{\left(n_x p'_x + n_y p'_y + 1\right)(n_z - 1)}$$

$$p_y = \frac{(n_y^2 + n_z - 1)\,p'_y + n_x n_y p'_x}{\left(n_x p'_x + n_y p'_y + 1\right)(n_z - 1)}$$



O'Sullivan and Stec "Sensor tilt via conic sections" IMVIP *141-144* (2020).

Lens Undistortion

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

Given the set of homogeneous co-ordinates $\hat{p}$, we seek to unproject these points to the surface of the unit sphere.

- The radial distortion function of the lens $r(\theta)$ is known.
- The radial value of any point $\vec{p}$ is the length

$$r = \sqrt{p_x^2 + p_y^2}$$

- The corresponding field angle θ of each point is the inverse of the radial distortion function.
  $\rightarrow$ numerically this can be calculated with a 1d interpolation.

Lens Undistortion

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

Given the set of homogeneous co-ordinates $\hat{p}$, we seek to unproject these points to the surface of the unit sphere.

- The azimuths are

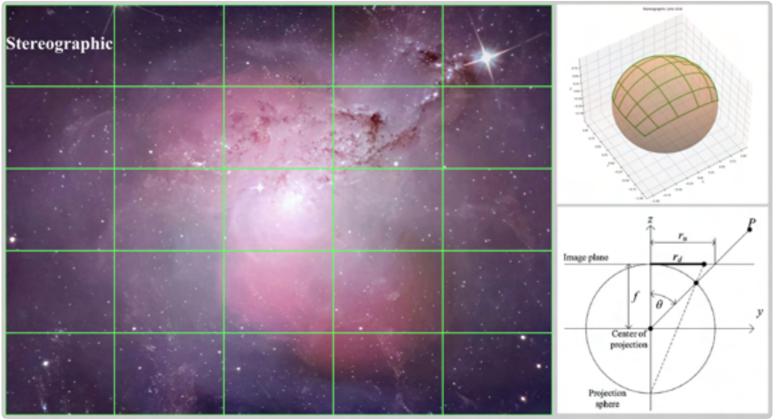
$$\left(\cos\left(\phi\right), \sin\left(\phi\right)\right) = \left(\frac{p_x}{r}, \frac{p_y}{r}\right)$$

- The spherical co-ordinates of each point is subsequently

$$\hat{P} = \begin{pmatrix} \sin\left(\theta\right)\cos\left(\phi\right) \\ \sin\left(\theta\right)\sin\left(\phi\right) \\ \cos\left(\theta\right) \end{pmatrix}$$

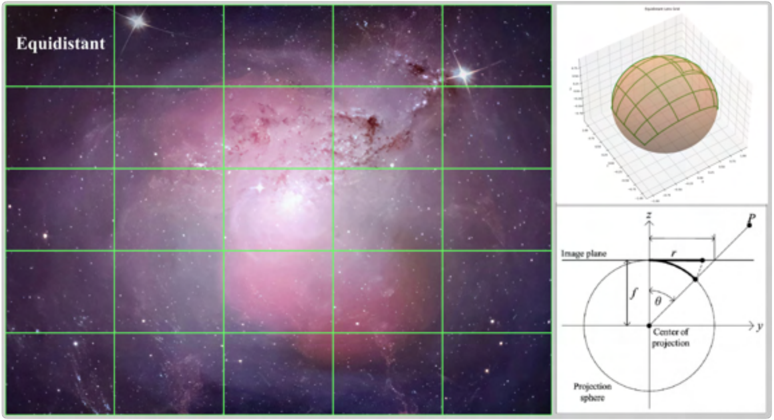
Lens Undistortion

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$



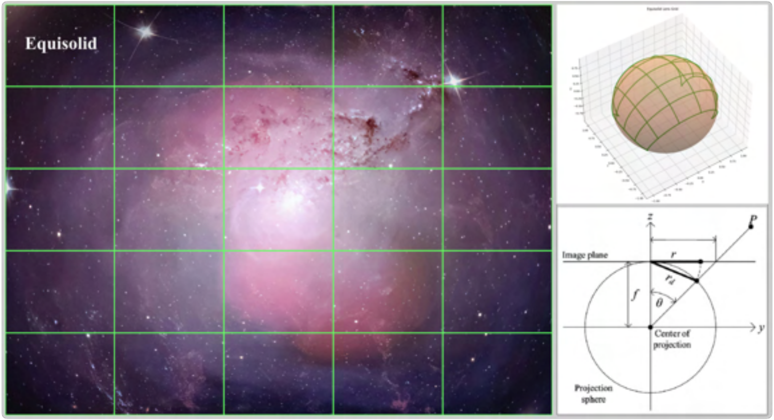
Lens Undistortion

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$



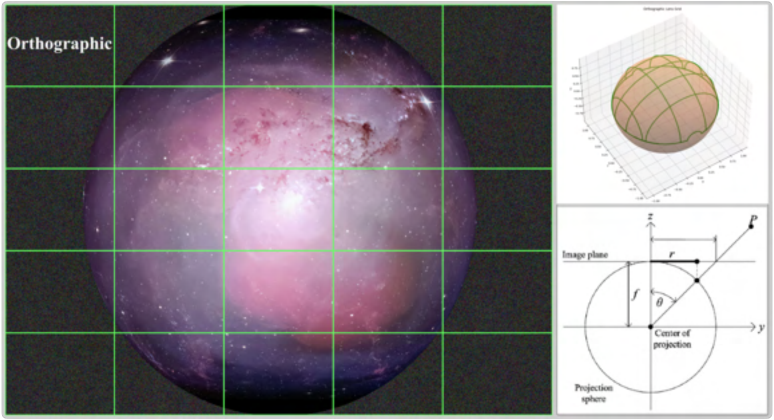
Lens Undistortion

$$\hat{x} \xmapsto{\text{intrinsics}} \hat{p}' \xmapsto{\text{sensor untilt}} \hat{p} \xmapsto{\text{lens undistortion}} \hat{P} \xmapsto{\text{extrinsics}} \hat{X}$$



Lens Undistortion

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$



Ray Tracing

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

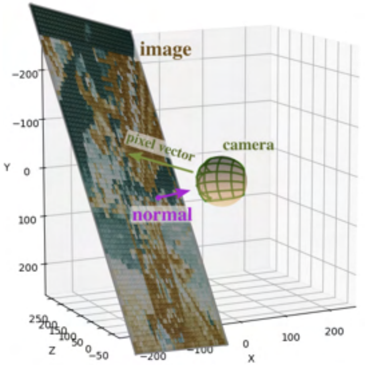
Line-Plane intersection:

1. The unit normal $\vec{n}_0 = (0, 0, -1)$, and plane center $\vec{u}_0 = (0, 0, 0)$, are transformed via the extrinsics as

$$\vec{n} = \hat{R}\vec{n}_0$$
$$\vec{u} = \vec{t} + \vec{u}_0$$

2. A plane is defined as the set of points $\hat{P}$ for which,

$$(\hat{P} - \vec{u}) \cdot \vec{n} = 0$$

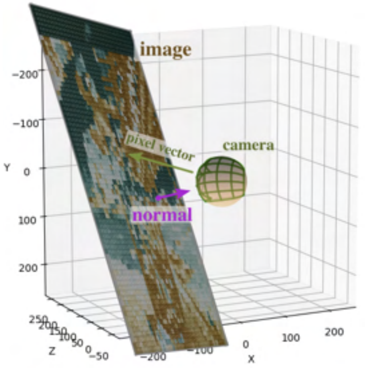


Ray Tracing

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

The unit pixel vectors are $\hat{\rho}$.
Each vector $\vec{\rho}_i$ is scaled by l_i to intersect the plane at $\vec{P}_i$.
For all $\vec{n} \cdot \vec{\rho}_i < 0$

$$\vec{P}_i = l_i \vec{\rho}_i$$



Plugging in we find the scaling

$$\left(\vec{P}_i - \vec{u}_i\right) \cdot \vec{n} = 0$$
$$\left(l_i \vec{\rho}_i - \vec{u}_i\right) \cdot \vec{n} = 0$$
$$l_i = \frac{\vec{u}_i \cdot \vec{n}}{\vec{\rho}_i \cdot \vec{n}}$$

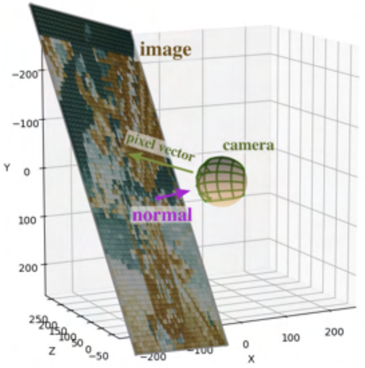
Ray Tracing

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p}' \xrightarrow{\text{sensor untilt}} \hat{p} \xrightarrow{\text{lens undistortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

Points are translated to the origin

$$\vec{X} = \hat{R}^t \left(l_i \vec{\rho}_i - \vec{t} \right)$$

Those $\vec{X}$ within the image bounds, are interpolated to find the RGB values.



Lens Remapping

Rectilinear remapping

- Sensor of (height, width) = (h, w).
- Create an array of $n = h * w$ pixel coordinates.

$$\hat{x} = \begin{pmatrix} x_0 & x_1 & \cdots & x_i & \cdots & x_{n-1} \\ y_0 & y_1 & \cdots & y_i & \cdots & y_{n-1} \\ 1 & 1 & \cdots & 1 & \cdots & 1 \end{pmatrix}$$

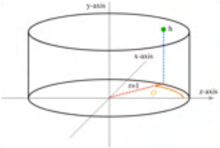
- Transform the camera coordinates

$$\hat{p}' = \hat{K}^{-1}\hat{x}$$

- Normalize to the unit sphere $\hat{p}' \mapsto \hat{\rho}$
- Project through lens model and interpolate image.

$$\hat{\rho} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

Lens Remapping



Cylindrical remapping

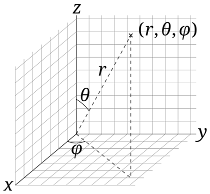
- Create cylindrical mesh grid in $(\hat{\phi}, \hat{h})$, of dimensions (h, w).
- Cast in cylindrical coordinates

$$\hat{\rho} = (x_i, y_i, z_i) = (\sin(\phi_i), h_i, \cos(\phi_i))$$

- Project through lens model and interpolate image.

$$\hat{\rho} \xmapsto{\text{lens distortion}} \hat{p} \xmapsto{\text{sensor tilt}} \hat{p}' \xmapsto{\text{intrinsics}} \hat{x}$$

Lens Remapping



Spherical remapping

- Create mesh grid in $(\hat{\theta}, \hat{\phi})$, of dimensions (h, w).
- Cast in spherical coordinates $\hat{\rho}$

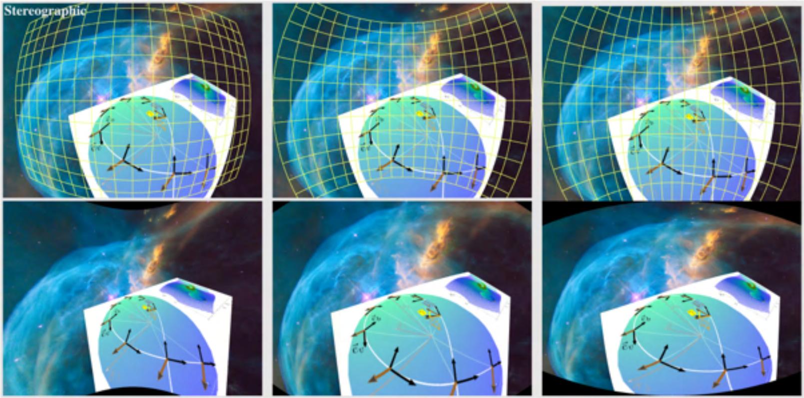
$$\rho_i = \left(x_i, y_i, z_i\right) = \left(\sin(\theta_i)\cos\left(\phi_i\right), \sin(\theta_i)\sin\left(\phi_i\right), \cos(\theta_i)\right)$$

- Project through lens model and interpolate image.

$$\hat{\rho} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{sensor tilt}} \hat{p}' \xrightarrow{\text{intrinsics}} \hat{x}$$

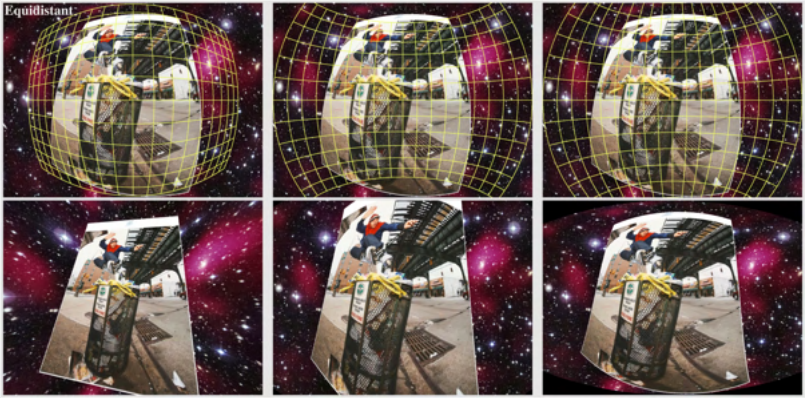
Lens Remapping

Stereographic



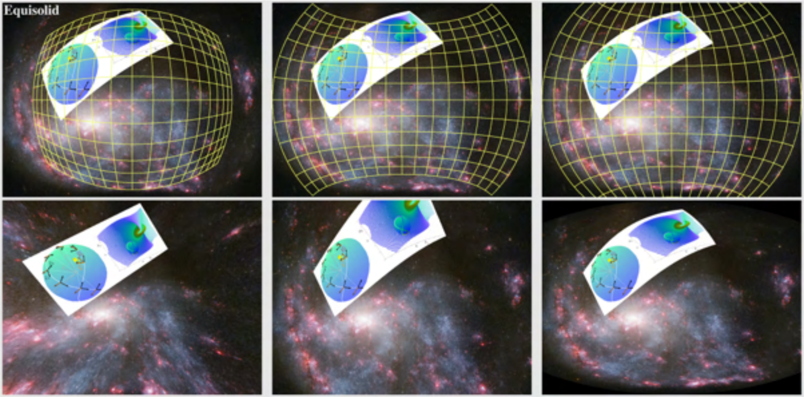
Lens Remapping

Equidistant



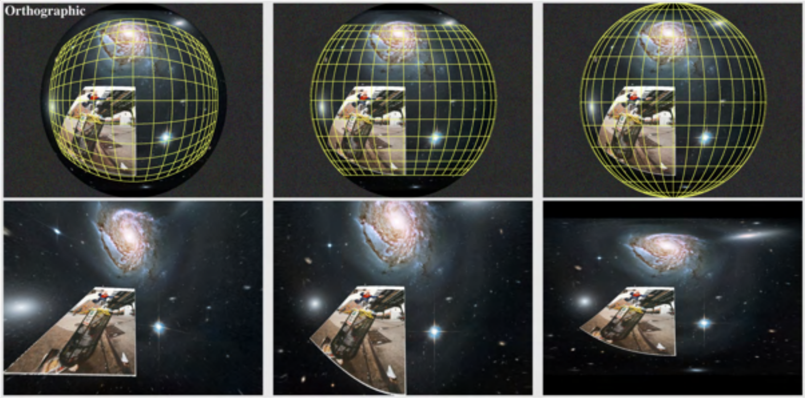
Lens Remapping

Equisolid



Lens Remapping

Orthographic



Section Outline

`Camera Calibration and Non-Linear Optimization`

Section Outline

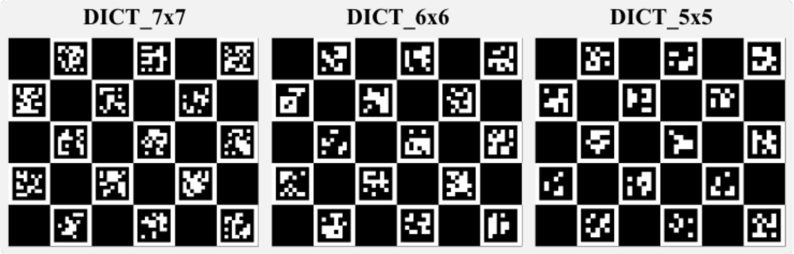
Camera calibration is the process of determining the lens distortion and intrinsics parameters, of a fixed focus camera.

The general method of camera calibration involves:

1. Capturing images of a target image, whose dimensions in the real world are known.
2. Adjusting the parameters of the camera model until the reprojected image from the model matches the target image.

Section Outline

`ChArUco` is an `opencv` library that generates chessboard style images with `ArUco` markers.

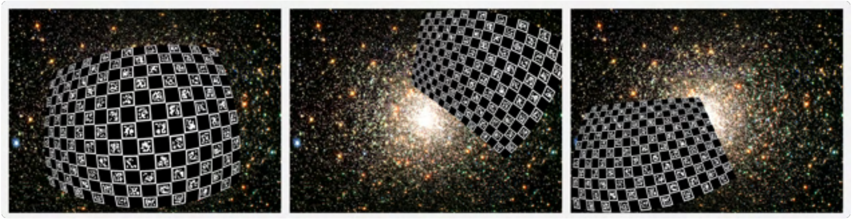


The library includes a corner detection algorithm that will detect and output the pixel coordinates of the chessboard corners with an average accuracy of about 0.5 pixels.

Section Outline

Camera Calibration procedure:

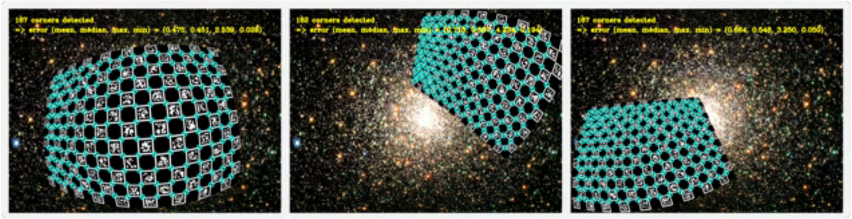
- `ChArUco` images are printed, the real world square size *in millimetres* is measured $\rightarrow$ initial positions $\hat{X}$ are known.
- Capture photographs of the `ChArUco` board in different positions $\rightarrow$ corner detection algorithm returns $\hat{x}'$.
- (1) Initialize parameters and calculate reprojections $\hat{X} \mapsto \hat{x}$.
- (2) Optimize parameters until $\hat{x}$ matches detected points $\hat{x}'$.



Section Outline

Camera Calibration procedure:

- `ChArUco` images are printed, the real world square size *in millimetres* is measured $\rightarrow$ initial positions $\hat{X}$ are known.
- Capture photographs of the `ChArUco` board in different positions $\rightarrow$ corner detection algorithm returns $\hat{x}'$.
- (1) Initialize parameters and calculate reprojections $\hat{X} \mapsto \hat{x}$.
- (2) Optimize parameters until $\hat{x}$ matches detected points $\hat{x}'$.



Model Initialization

`Model Initialization`

Model Initialization

Given a set of detected points $\hat{x}'_i$, for images $i = 1, 2, 3, \ldots$
We are required to estimate:

- $\hat{K}$ The camera matrix.
- $\hat{R}_i, \vec{t}_i$ The extrinsics for each image i.
- The lens distortion parameters.

Since the lens type is unknown, we use a polynomial to model the lens distortion, and optimize for the coefficients of the polynomial.

$$\vec{r}(\theta) = \theta + \kappa_2\theta^2 + \kappa_3\theta^3 + \kappa_4\theta^4 + \cdots$$

In this example we truncate the polynomial to the 4^{th} order. Hence we are required to determine $\kappa_2, \kappa_3, \kappa_4$. The first coefficient is always $\kappa_1 = 1$.

Model Initialization

Step #1 we ignore lens distortion.
Group the detected points per

$$\hat{x}'_i \approx \hat{K} \left[\hat{R}_i \mid \vec{t}_i \right] \hat{X}_i$$

The initial positions $\hat{X}_i$ are the same for all images i, hence this group of points is repeated (and transformed) for each image i. The array $\hat{X}$ contains the initial positions of the `ChArUco` target corners. Since the panel is initialized in the $z = 0$ plane, then

$$\hat{X}_i = \begin{pmatrix} X_0 & X_1 & \cdots & X_j & \cdots \\ Y_0 & Y_1 & \cdots & Y_j & \cdots \\ 0 & 0 & \cdots & 0 & \cdots \\ 1 & 1 & \cdots & 1 & \cdots \end{pmatrix}_i = \begin{pmatrix} \vec{X} \\ \vec{Y} \\ \vec{0} \\ \vec{1} \end{pmatrix}_i$$

with $\vec{X} = (X_0, X_1, \ldots, X_j, \ldots)$, $\vec{Y} = (Y_0, Y_1, \ldots, Y_j, \ldots)$.

Model Initialization

The sequence of mappings considered for model initialization is

$$\hat{X} \xmapsto{\text{extrinsics}} \hat{P} \xmapsto{\text{intrinsics}} \hat{P}' \xmapsto{\text{perspective}} \hat{x}'$$

The rotation matrix is expanded

$$\hat{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} = \begin{pmatrix} \vec{R}_1 & \vec{R}_2 & \vec{R}_3 \end{pmatrix}$$

Since $(Z_j)_i = 0$, consequently for each image i we have

$$\begin{pmatrix} \vec{P}'_x \\ \vec{P}'_y \\ \vec{P}'_z \end{pmatrix}_i = \begin{pmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} R_{11} & R_{12} & t_x \\ R_{21} & R_{22} & t_y \\ R_{31} & R_{32} & t_z \end{bmatrix}_i \begin{pmatrix} \vec{X} \\ \vec{Y} \\ \vec{1} \end{pmatrix}_i$$

Model Initialization

Let $\hat{h}_i = \hat{K}\left[\,\hat{R} \mid \vec{t}\,\right]_i$, therefore

$$\begin{pmatrix} \vec{P}'_x \\ \vec{P}'_y \\ \vec{P}'_z \end{pmatrix}_i = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}_i \begin{pmatrix} \vec{X} \\ \vec{Y} \\ \vec{1} \end{pmatrix}_i$$

The perspective projection is $\hat{P}' \xmapsto{\text{perspective}} \hat{x}'$:

$$\left(\vec{x}'\right)_i = \left(\frac{\vec{P}'_x}{\vec{P}'_z}\right)_i = \left(\frac{h_{11}\vec{X} + h_{12}\vec{Y} + h_{13}}{h_{31}\vec{X} + h_{32}\vec{Y} + h_{33}}\right)_i$$

$$\left(\vec{y}'\right)_i = \left(\frac{\vec{P}'_y}{\vec{P}'_z}\right)_i = \left(\frac{h_{21}\vec{X} + h_{22}\vec{Y} + h_{23}}{h_{31}\vec{X} + h_{32}\vec{Y} + h_{33}}\right)_i$$

Model Initialization

Rearrange the equations

$$
\begin{aligned}
h_{11}\vec{X} + h_{12}\vec{Y} + h_{13} - h_{31}\vec{x}'\vec{X} - h_{32}\vec{x}'\vec{Y} - h_{33}\vec{x}' &= 0 \\
h_{21}\vec{X} + h_{22}\vec{Y} + h_{23} - h_{31}\vec{y}'\vec{X} - h_{32}\vec{y}'\vec{Y} - h_{33}\vec{y}' &= 0
\end{aligned}
$$

Construct the matrix form:

$$
\begin{pmatrix}
\vec{X} & \vec{Y} & \vec{1} & 0 & 0 & 0 & -\vec{x}'\vec{X} & -\vec{x}'\vec{Y} & -\vec{x}' \\
0 & 0 & 0 & \vec{X} & \vec{Y} & \vec{1} & -\vec{y}'\vec{X} & -\vec{y}'\vec{Y} & -\vec{y}'
\end{pmatrix}_i
\begin{pmatrix}
h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33}
\end{pmatrix}_i
= 0
$$

Model Initialization

Since the equation is invariant up to a scale factor we divide by h_{33}, such that all elements are now $h_{uv} = h_{uv}/h_{33}$. Rearrange to find

$$\begin{pmatrix} \vec{X} & \vec{Y} & \vec{1} & 0 & 0 & 0 & -\vec{x}'\vec{X} & -\vec{x}'\vec{Y} \\ 0 & 0 & 0 & \vec{X} & \vec{Y} & \vec{1} & -\vec{y}'\vec{X} & -\vec{y}'\vec{Y} \end{pmatrix}_i \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix}_i = \begin{pmatrix} \vec{x}' \\ \vec{y}' \end{pmatrix}_i$$

Therefore we have an equation of the form $\hat{M}\,\vec{h} = \vec{b}$, which can be solved via linear least squares.

Model Initialization

Linear Least squares:
Since $\hat{M}\,\vec{h} \approx \vec{b} \;\rightarrow\; \hat{M}^t\,\hat{M}\,\vec{h} \approx \hat{M}^t\vec{b}$, and multiplying across by the inverse we have

$$\vec{h} \;=\; \left(\hat{M}^t\,\hat{M}\right)^{-1}\hat{M}^t\vec{b}$$

The homography matrix

$$\hat{h} \;=\; \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \;=\; [\;\vec{h}_1\;\vec{h}_2\;\vec{h}_3\;]$$

relates to the planar projection matrix via the scale $a = h_{33}$

$$a\,[\;\vec{h}_1\;\vec{h}_2\;\vec{h}_3\;] \;=\; \hat{K}\,[\;\vec{R}_1\;\vec{R}_2\;\vec{t}\;]$$

Model Initialization: Zhang's Method

The proceeding follows Zhang's method for camera calibration. Consider that

$$\vec{R}_1 = a\,\hat{K}^{-1}\,\vec{h}_1 \qquad \vec{R}_2 = a\,\hat{K}^{-1}\,\vec{h}_2$$

Since the vectors $\vec{R}_1$ and $\vec{R}_2$ are orthonormal, then

$$\vec{R}_1^t\vec{R}_2 = 0 = \vec{h}_1^t\left(\hat{K}^{-t}\hat{K}^{-1}\right)\vec{h}_2$$

$$\vec{R}_1^t\vec{R}_1 - \vec{R}_2^t\vec{R}_2 = 0 = \vec{h}_1^t\left(\hat{K}^{-t}\hat{K}^{-1}\right)\vec{h}_1 - \vec{h}_2^t\left(\hat{K}^{-t}\hat{K}^{-1}\right)\vec{h}_2$$

We have $$\hat{K}^{-t}\hat{K}^{-1} = \begin{pmatrix} \frac{1}{f^2} & 0 & -\frac{o_x}{f^2} \\ 0 & \frac{1}{\alpha^2 f^2} & -\frac{o_y}{\alpha^2 f^2} \\ -\frac{o_x}{f^2} & -\frac{o_y}{\alpha^2 f^2} & \frac{\alpha^2 f^2+\alpha^2 o_x^2+o_y^2}{\alpha^2 f^2} \end{pmatrix} = \hat{B}$$

as described by @CyrillStachniss [link]

Model Initialization: Zhang's Method

Let $\hat{B} = \hat{K}^{-t}\hat{K}^{-1}$,

$$\hat{B} = \begin{pmatrix} B_1 & 0 & B_3 \\ 0 & B_2 & B_4 \\ B_3 & B_4 & B_5 \end{pmatrix} \qquad \vec{h}_1 = \begin{pmatrix} h_{11} \\ h_{21} \\ h_{31} \end{pmatrix} \qquad \vec{h}_2 = \begin{pmatrix} h_{12} \\ h_{22} \\ h_{32} \end{pmatrix}$$

Rewrite the conditions in matrix form

$$\vec{h}_1^t \, \hat{B} \, \vec{h}_2 = 0 \qquad \vec{h}_1^t \, \hat{B} \, \vec{h}_1 - \vec{h}_2^t \, \hat{B} \, \vec{h}_2 = 0$$

```
In[1]:= B := {{B1, 0, B3}, {0, B2, B4}, {B3, B4, B5}}
        h1 := {{h11}, {h21}, {h31}}
        h2 := {{h12}, {h22}, {h32}}
        Bvector := {B1, B2, B3, B4, B5}
        C1matrix := {h11 h12, h21 h22, h31 h12 + h11 h32, h31 h22 + h21 h32, h31 h32}
        C2matrix := {h11^2 - h12^2, h21^2 - h22^2, 2 (h11 h31 - h12 h32), 2 (h21 h31 - h22 h32), h31^2 - h32^2}

In[7]:= Transpose[h1].B.h2 - Transpose[Bvector].C1matrix // Simplify
        Transpose[h1].B.h1 - Transpose[h2].B.h2 - Transpose[Bvector].C2matrix // Simplify

Out[7]= {{0}}

Out[8]= {{0}}
```

Model Initialization: Zhang's Method

Matrix Form:

$$\begin{pmatrix} h_{11}h_{12} & h_{21}h_{22} & h_{31}h_{12}+h_{11}h_{32} & h_{31}h_{22}+h_{21}h_{32} & h_{31}h_{32} \\ h_{11}^2-h_{12}^2 & h_{21}^2-h_{22}^2 & 2(h_{11}h_{31}-h_{12}h_{32}) & 2(h_{21}h_{31}-h_{22}h_{32}) & h_{31}^2-h_{32}^2 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{pmatrix} = 0$$

Construct the Linear Least Squares equation by setting $b_j = B_j / B_5$.

$$\hat{A}_i = \begin{pmatrix} h_{11}h_{12} & h_{21}h_{22} & h_{12}h_{31}+h_{11}h_{32} & h_{22}h_{31}+h_{21}h_{32} \\ h_{11}^2-h_{12}^2 & h_{21}^2-h_{22}^2 & 2(h_{11}h_{31}-h_{12}h_{32}) & 2(h_{21}h_{31}-h_{22}h_{32}) \end{pmatrix}_i$$

$$\vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \qquad \vec{C}_i = \begin{pmatrix} h_{31}h_{32} \\ h_{31}^2-h_{32}^2 \end{pmatrix}_i$$

For each image i.

Model Initialization: Zhang's Method

For a set of images we formulate the simultaneous equation, solvable by LLS

$$\hat{\Lambda}\,\vec{b} = -\vec{\Sigma} \qquad \rightarrow \qquad \vec{b} = -\left(\hat{\Lambda}^t\hat{\Lambda}\right)^{-1}\hat{\Lambda}^t\,\vec{\Sigma}$$

where

$$\hat{\Lambda} = \begin{pmatrix} \hat{A}_1 \\ \hat{A}_2 \\ \vdots \\ \hat{A}_i \\ \vdots \end{pmatrix} \qquad \vec{\Sigma} = \begin{pmatrix} \vec{C}_1 \\ \vec{C}_2 \\ \vdots \\ \vec{C}_i \\ \vdots \end{pmatrix}$$

Model Initialization: Zhang's Method

Given $\vec{b}$ we form the positive definite matrix

$$\hat{b} = \begin{pmatrix} b_1 & 0 & b_3 \\ 0 & b_2 & b_4 \\ b_3 & b_4 & 1 \end{pmatrix}$$

Perform the Cholesky decomposition of $\hat{b}$ into the product of a lower triangular matrix and its conjugate transpose,

$$\hat{b} = \hat{L}\,\hat{L}^t$$

Since the last element of $\hat{K}^{-1}$ is 1, we rescale $\hat{L}$ by dividing by it's last element and then solve for the intrinsic parameters.

Model Initialization: Zhang's Method

Previously we had the relations for the vectors of the rotation matrix

$$\vec{R}_1 = a\,\hat{K}^{-1}\,\vec{h}_1 \qquad \vec{R}_2 = a\,\hat{K}^{-1}\,\vec{h}_2$$

Set $a = 1$, and normalize each vector. The third vector is the normalized cross product.

$$\vec{R}_1 = \frac{\vec{R}_1}{|\vec{R}_1|} \qquad \vec{R}_2 = \frac{\vec{R}_2}{|\vec{R}_2|} \qquad \vec{R}_3 = \pm\frac{\vec{R}_1 \times \vec{R}_2}{|\vec{R}_1 \times \vec{R}_2|}$$

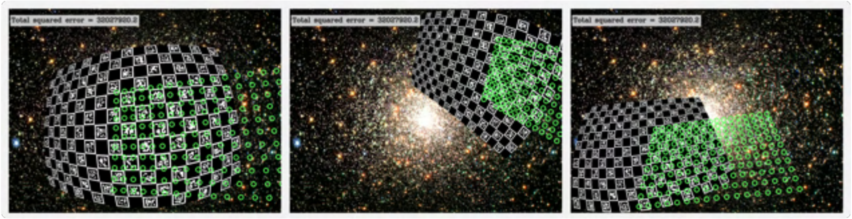
Hence we have either $\hat{R} = \begin{pmatrix} \vec{R}_1 & \vec{R}_2 & \pm\vec{R}_3 \end{pmatrix}$

To ensure orthornormality, convert the rotation matrix to a quaternion and back. The translation vector is initialized as using the mean scaling of $|\vec{R}_1|$, $|\vec{R}_2|$.

$$\vec{t} = \frac{2}{|\vec{R}_1| + |\vec{R}_2|}\hat{K}^{-1}\,\vec{h}_3$$

Accuracy and Alternatives

With the current version of the python code implementing Zhang's method, we obtain an initialization accuracy shown below.



Good initialization is essential for successful optimization, therefore improvements are needed, and one alternative is explored here.

Accuracy and Alternatives

A simplification

- Assume the optical center is the image center for initialization.
- Transform detected points $\hat{x}'$ accordingly.
- For initialization we estimate only the focal lengths $(f, \alpha f)$ in the camera matrix $\hat{K}$.

Since $\hat{B} = \hat{K}^{-t}\hat{K}^{-1}$,

$$\hat{B} = \begin{pmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_5 \end{pmatrix}$$

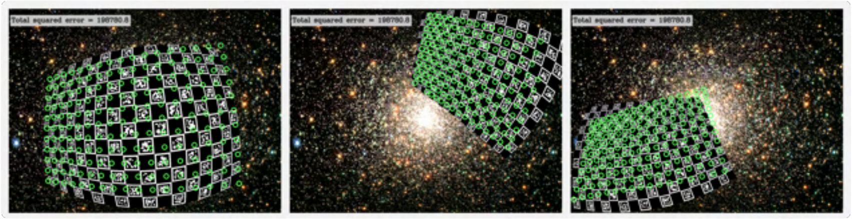
The homography relation simplifies to:

$$\begin{pmatrix} h_{11}h_{12} & h_{21}h_{22} & h_{31}h_{32} \\ h_{11}^2 - h_{12}^2 & h_{21}^2 - h_{22}^2 & h_{31}^2 - h_{32}^2 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ B_5 \end{pmatrix} = 0$$

Accuracy and Alternatives

The calculation of the camera matrix, and extrinsics proceeds as before.

The initialization results are slightly improved



The Objective Function

`Model Optimization`

The Objective Function

Given a set of initialized camera parameters $\vec{\eta}$, calculate the reprojected points via the mapping:

$$\hat{X} \xrightarrow{\text{extrinsics}} \hat{P} \xrightarrow{\text{lens distortion}} \hat{p} \xrightarrow{\text{intrinsics}} \hat{x}$$

The reprojection error is the squared distance between the detected points $\hat{x}'$, and the reprojected points $\hat{x}$.

$$\vec{\mathscr{D}} = \frac{1}{2}\left(\hat{x} - \hat{x}'\right) \cdot \left(\hat{x} - \hat{x}'\right)$$

$\vec{\mathscr{D}}$ is a one dimensional vector of length n (for n points) whereas the vector of deltas is $2 \times n$ dimensional.

$$\hat{x} - \hat{x}' = \begin{pmatrix} \delta\vec{x} & \delta\vec{y} \end{pmatrix}$$

Hence $\vec{\mathscr{D}} = \frac{1}{2}\left(\delta\vec{x} \cdot \delta\vec{x} + \delta\vec{y} \cdot \delta\vec{y}\right)$

The Objective Function

The total error is the dot product of the squared errors (sum of squares)

$$\mathscr{F} = \vec{\mathscr{D}} \cdot \vec{\mathscr{D}}$$

As the camera parameters $\vec{\eta}$ are optimized the squared errors $\vec{\mathscr{D}}(\vec{\eta})$ approach a minimum.
We seek to minimize the squared reprojection error for all points in the vector $\vec{\mathscr{D}}$. Consequently $\vec{\mathscr{D}}$ is our objective function.

Consider the Taylor series expansion of the function $f(x)$ around a.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

The Objective Function

By analogy we expand the objective function

$$\vec{\mathscr{D}}(\vec{\eta}_{j+1}) = \vec{\mathscr{D}}(\vec{\eta}_j) + \nabla\vec{\mathscr{D}}(\vec{\eta}_j)(\vec{\eta}_{j+1} - \vec{\eta}_j) + \nabla^2\vec{\mathscr{D}}(\vec{\eta}_j)(\vec{\eta}_{j+1} - \vec{\eta}_j)^2 + \dots$$

Take the Taylor series to first order such that $\vec{\mathscr{D}}(\vec{\eta}_{j+1}) \approx 0$, and rearrange

$$\nabla\vec{\mathscr{D}}(\vec{\eta}_j)(\vec{\eta}_{j+1} - \vec{\eta}_j) = -\vec{\mathscr{D}}(\vec{\eta}_j)$$

$\nabla\vec{\mathscr{D}}$ is the Jacobian denoted $\mathscr{J}$. Multiply across both sides by $\mathscr{J}^t(\eta_j)$, and take the inverse to find

$$\vec{\eta}_{j+1} = \vec{\eta}_j - \left(\mathscr{J}(\eta_j)^t\mathscr{J}(\eta_j)\right)^{-1}\mathscr{J}(\eta_j)^t\vec{\mathscr{D}}(\vec{\eta}_j)$$

This is the Gauss-Newton optimization algorithm, used to solve non-linear least squares problems, which is equivalent to minimizing a sum of squared function values.

The Jacobian

The Jacobian is a matrix of partial derivatives, of dimensions $n \times m$, for n detected points and m parameters.

$$\hat{\mathscr{J}} = \begin{pmatrix} \partial_{\eta_1}\vec{\mathscr{D}} & \partial_{\eta_2}\vec{\mathscr{D}} & \partial_{\eta_3}\vec{\mathscr{D}} & \cdots & \partial_{\eta_j}\vec{\mathscr{D}} & \cdots & \partial_{\eta_m}\vec{\mathscr{D}} \end{pmatrix}$$

In this analysis we are not optimizing for sensor tilt. Hence the Jacobian contains 3 groups of derivatives that we need consider:

$$\underbrace{\partial_{\eta_1}\vec{\mathscr{D}}\ \partial_{\eta_2}\vec{\mathscr{D}}\ \partial_{\eta_3}\vec{\mathscr{D}}\ \partial_{\eta_4}\vec{\mathscr{D}}}_{\text{intrinsics}} \qquad \underbrace{\partial_{\eta_5}\vec{\mathscr{D}}\ \partial_{\eta_6}\vec{\mathscr{D}}\ \partial_{\eta_7}\vec{\mathscr{D}}}_{\text{lens}}$$

$$\underbrace{\partial_{\eta_8}\vec{\mathscr{D}}\ \partial_{\eta_9}\vec{\mathscr{D}} \cdots \partial_{\eta_{13}}\vec{\mathscr{D}}}_{\text{extrinsics}}{}^{i=1,\ldots,n}$$

$$\vec{\eta} = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \\ \eta_6 \\ \eta_7 \\ \eta_8 \\ \eta_9 \\ \eta_{10} \\ \eta_{11} \\ \eta_{12} \\ \eta_{13} \\ \vdots \end{pmatrix} = \begin{pmatrix} f \\ \alpha \\ dX \\ dY \\ \kappa_2 \\ \kappa_3 \\ \kappa_4 \\ \psi_r \\ \psi_p \\ \psi_y \\ t_x \\ t_y \\ t_z \\ \vdots \end{pmatrix}$$

The Jacobian

The projected points $\hat{x}$, follow the sequence of operations from the forward mapping (with sensor tilt omitted):

$$\hat{X} \xmapsto{\text{extrinsics}} \hat{P} \xmapsto{\text{lens distortion}} \hat{p} \xmapsto{\text{intrinsics}} \hat{x}$$

The mapping is our guideline when calculating the derivatives of the Jacobian.

$$\partial_a \vec{\mathscr{D}} = \frac{1}{2}\left(\hat{x} - \hat{x}'\right) \cdot \partial_a \hat{x}$$

When taking the derivatives of $\hat{x}$, we need to be mindful that the derivatives are taken in the order from right to left. Consequently we reverse the written order of the mapping, but retain knowledge that it refers to the forward mapping.

$$\hat{x} \xmapsto{\text{intrinsics}} \hat{p} \xmapsto{\text{lens distortion}} \hat{P} \xmapsto{\text{extrinsics}} \hat{X}$$

The Jacobian

$$\color{blue}{\hat{x} \xmapsto{\text{intrinsics}}} \hat{p} \xmapsto{\text{lens distortion}} \hat{P} \xmapsto{\text{extrinsics}} \hat{X}$$

The first group of derivatives we need consider are the intrinsics.

$$\underbrace{\partial_{\eta_1}\vec{\mathscr{D}} \qquad \partial_{\eta_2}\vec{\mathscr{D}} \qquad \partial_{\eta_3}\vec{\mathscr{D}} \qquad \partial_{\eta_4}\vec{\mathscr{D}}}_{\text{intrinsics}}$$

Since $\hat{x} = \hat{K}\hat{p}$, then the partials are simply $\partial_{\eta_a}\hat{x} = \left(\partial_{\eta_a}\hat{K}\right)\hat{p}$, with

$$\hat{K} = \begin{pmatrix} f & 0 & x_c + dX \\ 0 & \alpha f & y_c + dY \\ 0 & 0 & 1 \end{pmatrix} \qquad \partial_{\eta_1}\hat{K} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \partial_{\eta_2}\hat{K} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\partial_{\eta_3}\hat{K} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \partial_{\eta_4}\hat{K} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

The Jacobian

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p} \xrightarrow{\text{lens distortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

The second group of derivatives we need consider are the lens distortion parameters.

$$\overbrace{\partial_{\eta_5}\vec{\mathscr{D}} \qquad \partial_{\eta_6}\vec{\mathscr{D}} \qquad \partial_{\eta_7}\vec{\mathscr{D}}}_{\text{lens distortion}}$$

Since the lens distorted co-ordinates are

$$\hat{p} = \begin{pmatrix} r(\theta)\cos(\phi) \\ r(\theta)\sin(\phi) \\ 1 \end{pmatrix} \qquad \text{with} \qquad r(\theta) = \theta + \kappa_2\theta^2 + \kappa_3\theta^3 + \kappa_4\theta^4$$

The Jacobian

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p} \xrightarrow{\text{lens distortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

The second group of derivatives we need consider are the lens distortion parameters.

$$\overbrace{\partial_{\eta_5}\vec{\mathscr{D}} \qquad \partial_{\eta_6}\vec{\mathscr{D}} \qquad \partial_{\eta_7}\vec{\mathscr{D}}}$$

lens distortion

The derivates are given by

$$\partial_{\eta_j}\hat{p} = \begin{pmatrix} \partial_{\eta_j}\left[r(\theta)\right]\cos\left(\phi\right) \\ \partial_{\eta_j}\left[r(\theta)\right]\sin\left(\phi\right) \\ 0 \end{pmatrix} \quad \text{with} \quad \begin{aligned} \partial_{\eta_5} r(\theta) &= \theta^2 \\ \partial_{\eta_6} r(\theta) &= \theta^3 \\ \partial_{\eta_7} r(\theta) &= \theta^4 \end{aligned}$$

The Jacobian

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p} \xrightarrow{\text{lens distortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

The third group we need consider are the extrinsic parameters.

$$\overbrace{\partial_{\eta_8}\vec{\mathscr{D}} \quad \partial_{\eta_9}\vec{\mathscr{D}} \quad \underset{\text{extrinsics}}{\partial_{\eta_{10}}\vec{\mathscr{D}} \quad \partial_{\eta_{11}}\vec{\mathscr{D}}} \quad \partial_{\eta_{12}}\vec{\mathscr{D}} \quad \partial_{\eta_{13}}\vec{\mathscr{D}}}^{\text{for } i=1,2,..}$$

Recall that $r(\theta) = \theta + \kappa_2\theta^2 + \kappa_3\theta^3 + \kappa_4\theta^4$, and

$$\vec{p} = \begin{pmatrix} r(\theta)\cos(\phi) \\ r(\theta)\sin(\phi) \\ 1 \end{pmatrix} \qquad \theta = \tan^{-1}\left(\frac{\sqrt{P_x^2 + P_y^2}}{P_z}\right)$$

$$\vec{P} = [\,\hat{R} \mid \vec{t}\,]\,\vec{X} \qquad \phi = \tan^{-1}\left(\frac{P_y}{P_x}\right)$$

$$\partial_j \equiv (\eta_8, \eta_9, \eta_{10}, \eta_{11}, \eta_{12}, \eta_{13}) = (\psi_r, \psi_p, \psi_y, t_x, t_y, t_z)$$

The Jacobian

$$\hat{x} \xmapsto{\text{intrinsics}} \hat{p} \xmapsto{\text{lens distortion}} \hat{P} \xmapsto{\text{extrinsics}} \hat{X}$$

The third group we need consider are the extrinsic parameters.

$$\overbrace{\underbrace{\partial_{\eta_8}\vec{\mathscr{D}} \quad \partial_{\eta_9}\vec{\mathscr{D}} \quad \partial_{\eta_{10}}\vec{\mathscr{D}} \quad \partial_{\eta_{11}}\vec{\mathscr{D}} \quad \partial_{\eta_{12}}\vec{\mathscr{D}} \quad \partial_{\eta_{13}}\vec{\mathscr{D}}}_{\text{extrinsics}}}^{\text{for } i=1,2,..}$$

First consider

$$\partial_j \vec{p} = \begin{pmatrix} \partial_j [r(\theta)] \cos(\phi) - r(\theta) \sin(\phi) \partial_j \phi \\ \partial_j [r(\theta)] \sin(\phi) + r(\theta) \cos(\phi) \partial_j \phi \\ 0 \end{pmatrix}$$

$$\partial_j [r(\theta)] = \left(1 + 2\kappa_2 \theta + 3\kappa_3 \theta^2 + 4\kappa_4 \theta^3\right) \partial_j \theta$$

We are required to evaluate $\left(\partial_j \phi, \partial_j \theta\right)$.

The Jacobian

$$\hat{x} \xmapsto{\text{intrinsics}} \hat{p} \xmapsto{\text{lens distortion}} \hat{P} \xmapsto{\text{extrinsics}} \hat{X}$$

The third group we need consider are the extrinsic parameters.

$$\overbrace{\partial_{\eta_8}\vec{\mathscr{D}} \quad \partial_{\eta_9}\vec{\mathscr{D}} \quad \underset{\text{extrinsics}}{\partial_{\eta_{10}}\vec{\mathscr{D}} \quad \partial_{\eta_{11}}\vec{\mathscr{D}}} \quad \partial_{\eta_{12}}\vec{\mathscr{D}} \quad \partial_{\eta_{13}}\vec{\mathscr{D}}}^{\text{for } i=1,2,..}$$

The derivative of the arctan function is

$$\frac{\partial}{\partial_x}\left[\tan^{-1}\left(f(x)\right)\right] = \frac{1}{1+f(x)^2} f'(x)$$

Therefore

$$\partial_j \phi = \frac{1}{1+\left(\frac{P_y}{P_x}\right)^2}\left(\frac{\partial_j P_y}{P_x} - \frac{P_y \partial_j P_x}{P_x^2}\right) = \frac{P_x^2}{P_x^2+P_y^2}\frac{P_x \partial_j P_y - P_y \partial_j P_x}{P_x^2}$$

The Jacobian

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p} \xrightarrow{\text{lens distortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

The third group we need consider are the extrinsic parameters.

$$\overbrace{\underbrace{\partial_{\eta_8}\vec{\mathscr{D}} \quad \partial_{\eta_9}\vec{\mathscr{D}} \quad \partial_{\eta_{10}}\vec{\mathscr{D}} \quad \partial_{\eta_{11}}\vec{\mathscr{D}} \quad \partial_{\eta_{12}}\vec{\mathscr{D}} \quad \partial_{\eta_{13}}\vec{\mathscr{D}}}_{\text{extrinsics}}}^{\text{for } i=1,2,..}$$

The derivative of the arctan function is

$$\frac{\partial}{\partial_x}\left[\tan^{-1}\left(f(x)\right)\right] = \frac{1}{1+f(x)^2}f'(x)$$

Therefore

$$\partial_j \phi = \frac{P_x \partial_j P_y - P_y \partial_j P_x}{P_x^2 + P_y^2}$$

The Jacobian

$$\hat{x} \xmapsto{\text{intrinsics}} \hat{p} \xmapsto{\text{lens distortion}} \hat{P} \xmapsto{\text{extrinsics}} \hat{X}$$

The third group we need consider are the extrinsic parameters.

$$\underbrace{\overbrace{\partial_{\eta_8}\vec{\mathscr{D}} \quad \partial_{\eta_9}\vec{\mathscr{D}} \quad \partial_{\eta_{10}}\vec{\mathscr{D}} \quad \partial_{\eta_{11}}\vec{\mathscr{D}} \quad \partial_{\eta_{12}}\vec{\mathscr{D}} \quad \partial_{\eta_{13}}\vec{\mathscr{D}}}^{\text{for } i=1,2,..}}_{\text{extrinsics}}$$

And the derivative of arctan θ is

$$\partial_j\theta = \frac{P_z^2}{P_x^2+P_y^2+P_z^2}\partial_j\left[\frac{\sqrt{P_x^2+P_y^2}}{P_z}\right]$$

$$\partial_j\theta = \frac{P_z^2}{P_x^2+P_y^2+P_z^2}\left(\frac{P_x\partial_jP_x+P_y\partial_jP_y}{P_z\sqrt{P_x^2+P_y^2}} - \frac{\sqrt{P_x^2+P_y^2}}{P_z^2}\partial_jP_z\right)$$

The Jacobian

$$\hat{x} \xrightarrow{\text{intrinsics}} \hat{p} \xrightarrow{\text{lens distortion}} \hat{P} \xrightarrow{\text{extrinsics}} \hat{X}$$

The third group we need consider are the extrinsic parameters.

$$\underbrace{\overbrace{\partial_{\eta_8}\vec{\mathscr{D}} \quad \partial_{\eta_9}\vec{\mathscr{D}} \quad \partial_{\eta_{10}}\vec{\mathscr{D}} \quad \partial_{\eta_{11}}\vec{\mathscr{D}} \quad \partial_{\eta_{12}}\vec{\mathscr{D}} \quad \partial_{\eta_{13}}\vec{\mathscr{D}}}^{\text{for } i=1,2,..}}_{\text{extrinsics}}$$

Since $\vec{P} = [\,\hat{R} \,|\, \vec{t}\,]\,\hat{X}$,

$$\begin{aligned} \partial_{\psi_r}\vec{P} &= [\,\partial_{\psi_r}\hat{R} \,|\, \vec{0}\,]\,\hat{X} & \qquad \partial_{t_x}\vec{P} &= [\,\hat{0} \,|\, \partial_{t_x}\vec{t}\,]\,\hat{X} \\ \partial_{\psi_p}\vec{P} &= [\,\partial_{\psi_p}\hat{R} \,|\, \vec{0}\,]\,\hat{X} & \qquad \partial_{t_y}\vec{P} &= [\,\hat{0} \,|\, \partial_{t_y}\vec{t}\,]\,\hat{X} \\ \partial_{\psi_y}\vec{P} &= [\,\partial_{\psi_y}\hat{R} \,|\, \vec{0}\,]\,\hat{X} & \qquad \partial_{t_z}\vec{P} &= [\,\hat{0} \,|\, \partial_{t_z}\vec{t}\,]\,\hat{X} \end{aligned}$$

Gauss-Newton

`Gauss-Newton Regression Tests`

Gauss-Newton

From an initialized set of camera parameters η_0, the Gauss-Newton regression function updates the parameters as

$$\vec{\eta}'_{j+1} = \vec{\eta}_j - \left(\mathscr{J}(\eta_j)^t \mathscr{J}(\eta_j)\right)^{-1} \mathscr{J}(\eta_j)^t \vec{\mathscr{D}}(\vec{\eta}_j)$$

At each iteration the total squared error is calculated. Assuming a good initialization, the total squared error will converge to a minimum value.

However, at lower values the squared error can jump between high and low values, and can even move outside of the local minima region and increase substantially, or even lose convergence.

Gauss-Newton

From an initialized set of camera parameters η_0, the Gauss-Newton regression function updates the parameters as

$$\vec{\eta}'_{j+1} = \vec{\eta}_j - \left(\mathscr{J}(\eta_j)^t \mathscr{J}(\eta_j)\right)^{-1} \mathscr{J}(\eta_j)^t \vec{\mathscr{D}}(\vec{\eta}_j)$$

To combat the jumping of the squared error, and the loss of convergence, we apply two techniques.

- Update a record of the parameter vector for the lowest squared error.
- Apply a low pass filter that increases as the squared error decreases.

In the python code we have set the number of max iterations to $m_{\text{iter}} = 999$.

Gauss-Newton

From an initialized set of camera parameters η_0, the Gauss-Newton regression function updates the parameters as

$$\vec{\eta}'_{j+1} = \vec{\eta}_j - \left(\mathscr{J}(\eta_j)^t \mathscr{J}(\eta_j)\right)^{-1} \mathscr{J}(\eta_j)^t \vec{\mathscr{D}}(\vec{\eta}_j)$$

The low pass filter updates the parameter vector as

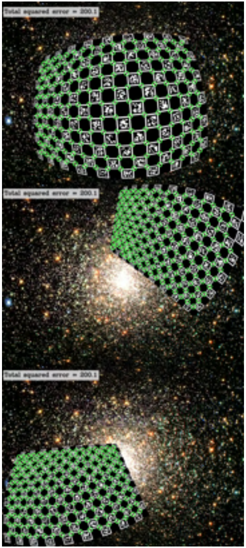
$$\vec{\eta}_{j+1} = \lambda \vec{\eta}_j + (1-\lambda)\vec{\eta}'_{j+1}$$

Initially $\lambda = 0.45$, and once the squared error drops below a threshold value of $a_{\text{trigger}} = 500$, the smoothing parameter increases as a function of the iteration counter c, and $\lambda \to 1$.

$$\lambda = 1.0 - 0.5 * ((m_{\text{iter}} - c) / m_{\text{iter}})^2$$

$\vec{\eta}_{j+1}$ is the updated parameter vector in the subsequent loop.

Gauss-Newton: Regression Tests



Calibration Results

Camera Matrix
focal length = 457.269
α = 1.049
dX = 6.285
dY = -2.662

Lens Distortion
k2 = 0.040406
k3 = -0.038533
k4 = 0.032426

Extrinsics
image00 panel01 roll = -6.863
image00 panel01 pitch = -39.990
image00 panel01 yaw = -24.350
image00 panel01 tX = -173.931
image00 panel01 tY = 84.151
image00 panel01 tZ = 200.478

---

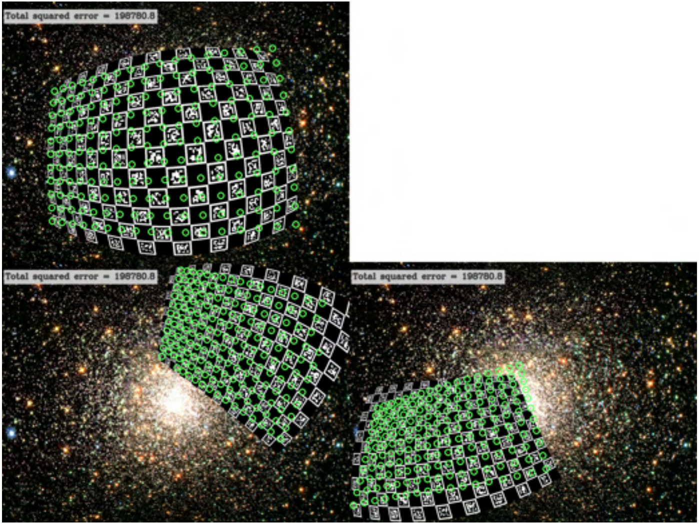
image01 panel01 roll = 33.548
image01 panel01 pitch = 14.934
image01 panel01 yaw = 49.900
image01 panel01 tX = 98.351
image01 panel01 tY = -114.767
image01 panel01 tZ = 232.418

---

image02 panel01 roll = -5.288
image02 panel01 pitch = -11.781
image02 panel01 yaw = 3.536
image02 panel01 tX = -41.312
image02 panel01 tY = 19.934
image02 panel01 tZ = 154.889

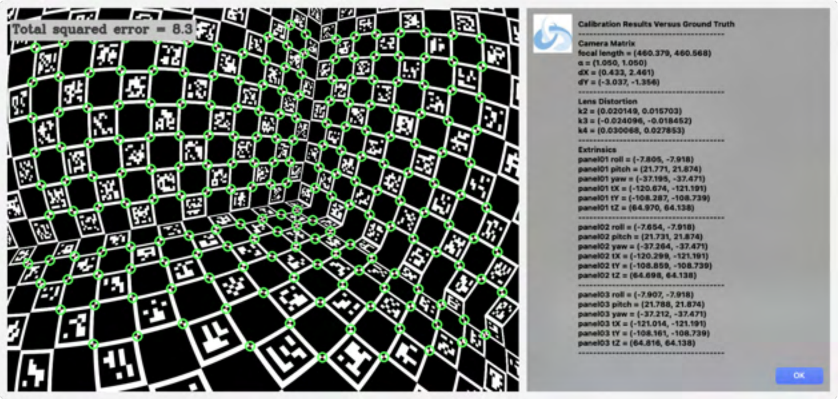
OK

Gauss-Newton: Regression Tests

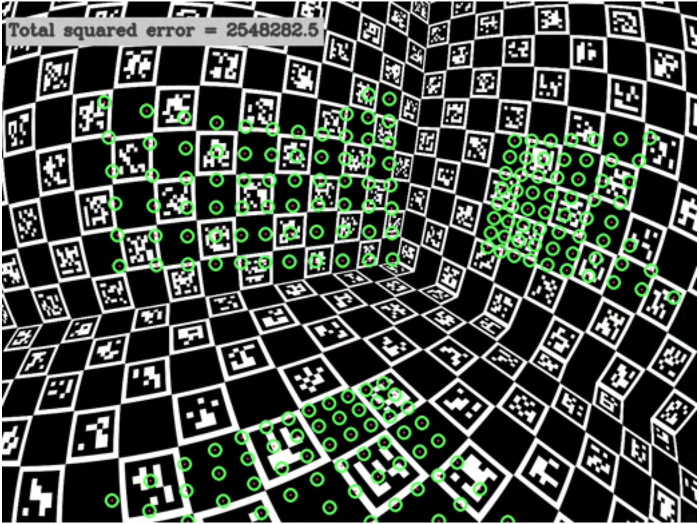


Gauss-Newton: Regression Tests

Calibration results for a 3 panel target:

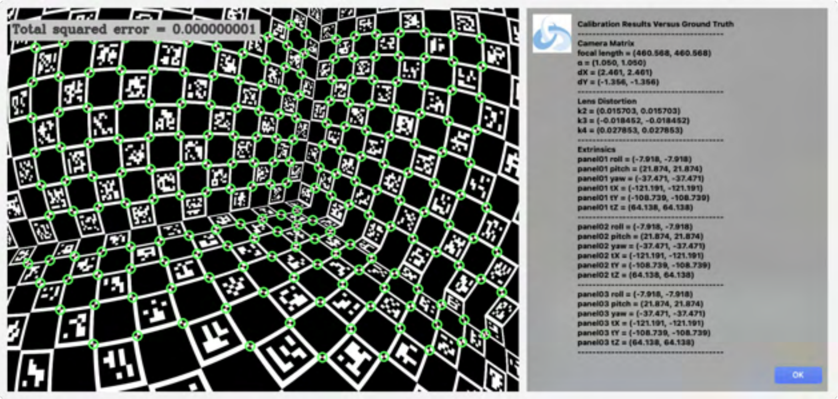


Gauss-Newton: Regression Tests

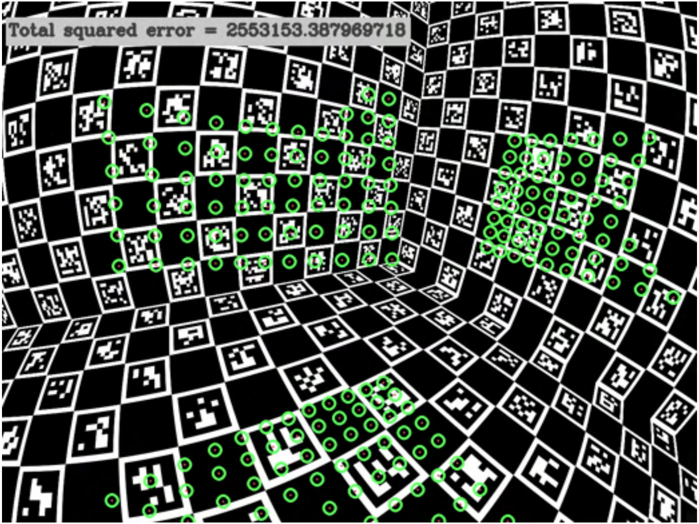


Gauss-Newton: Regression Tests

Calibration results for a 3 panel target with ground truth corners:



Gauss-Newton: Regression Tests



Conclusions and Outlook

Virtual Camera Calibration summary:

- Point Centric Camera Model with imager sensor tilt modelled via conic sections.
- Geometric Calibration using Zhang's Initialization method and Gauss-Newton optimization.
- Calibration accuracy is dependent on corner detection accuracy.

Improvements to virtual camera calibrator can be found in:

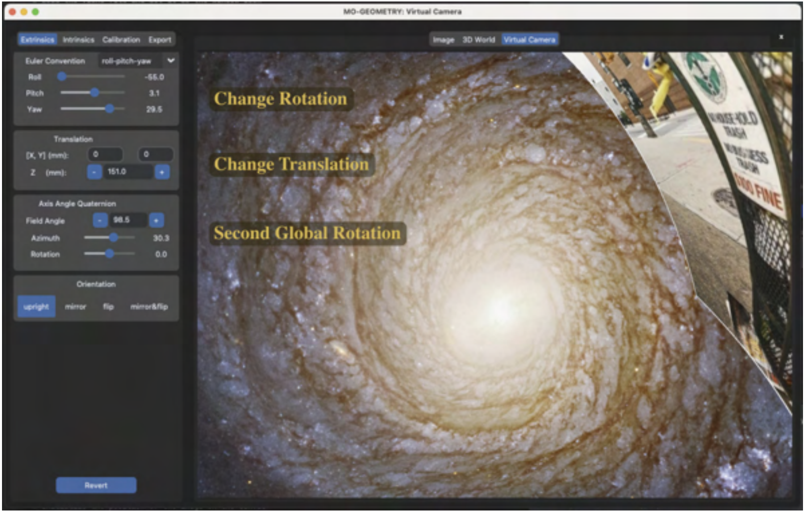
1. More accurate initialization.
2. Sensor tilt optimization.
3. Extend the ray tracing to a light source.

Conclusions and Outlook

Advancements beyond the point centric camera model may include the simulation and/or calibration of:

- Lateral and Longitudinal Chromatic Aberration
- Petzval Field Curvature
- Point Spread Function over field angle, wavelength.
- Moving entrance and exit pupils over wavelength.
- Defocus

Virtual Camera

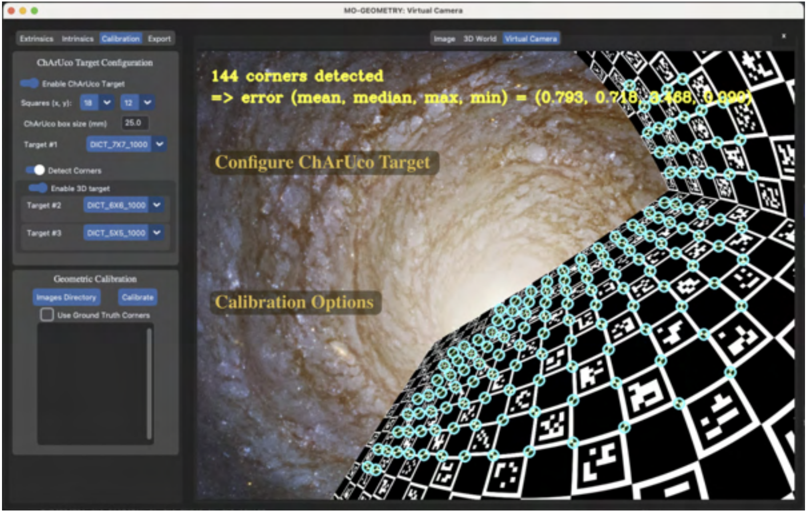


Code available on GitHub: mo-geometry/conic_sections

Virtual Camera

Code available on GitHub: mo-geometry/conic_sections

Virtual Camera



Code available on GitHub: mo-geometry/conic_sections

Virtual Camera

Thank you for your attention!