Chapter 1

Algorithm Description

The algorithm described in Chapter!!!Unresolved reference!!!3 was implemented in Python programming language. An open source machine vision library OpenCV was utilized to perform feature extraction and tracking. The feature extraction method used was the Shi-Tomasi corner detector. Feature tracking was accomplished through the pyramid implementation of Lucas-Kanade optical flow method .

1.1 Camera Centric Inverse Depth Parameterization

The standard way of describing a feature's position is to use the Euclidean XYZ parameterization. In practical outdoor range estimation problem, the algorithm must deal with features located at near infinity. Inverse depth parameterization overcomes this problem by representing range d in its inverse form $\rho = 1/d$. In addition, features at infinity contribute in estimating the camera rotational motion even though they offer little information on camera translational motion. Furthermore, inverse depth parameterization allows us to initialize features into the EKF framework before it is safely triangulated.

The inverse depth parameterization used in this work was first introduced in . All features and camera positions are referred to a world reference frame. When used in an Extended Kalman Filter framework, the system suffers decreasing linearity when camera is moving away from the world origin. A modified method which uses the camera center as the origin was proposed in . Our work has adopted the camera centered approach with minor modification to integrate the inertial measurements.

A scene point p_i^C can be defined by 6 parameters, with the superscript C representing a camera reference frame. ():

$$p_i^C = \begin{bmatrix} x_i^C & y_i^C & z_i^C & \rho_i & \varphi_i^C & \theta_i^C \end{bmatrix} \tag{1}$$

The first three parameters $[x_i^C, y_i^C, z_i^C]$ represent the initial position where the feature is first observed. ρ_i is the inverse distance from the initialization position to the feature. The elevation-azimuth pair $[\phi_i^C, \theta_i^C]$ encodes a unit vector pointing from the initialization point to the feature. The vector is given by

$$m(\phi_i^C, \theta_i^C) = \begin{bmatrix} \cos\varphi_i^C \cos\theta_i^C \\ \cos\varphi_i^C \sin\theta_i^C \\ \sin\varphi_i^C \end{bmatrix} \tag{2}$$

1.2 Modeling the System with Extended Kalman Filter

1.2.1 Full State Vector

The EKF state vector is defined as

$$x = \begin{bmatrix} OX_W^C & c^C & r^C & p_1^C & p_2^C & \dots \end{bmatrix} \tag{3}$$

where $OX_W^C = \begin{bmatrix} O_x^C & O_y^C & O_z^C & W_x^C & W_y^C & W_z^C \end{bmatrix}^T$ contains translation parameters $O_{x,y,z}^C$ and rotation parameters $W_{x,y,z}^C$ to transform the camera reference frame to the world reference frame. $\left(c^C, r^C\right)^T$ represents the camera translation and rotation motion frame by frame in Euclidean coordinates, and p_i^C contains the feature parameters as described in the previous section.

1.2.2 Prediction

(Will be updated to a more complete form)

For a prediction step at time k, the world frame and features parameters are kept unchanged from time $k-1$. The camera parameters are updated using the new inertial measurements: velocity v^C, acceleration a^C, and rate of change in roll/pitch/yaw w^C. The camera motion parameters at time k are then

$$\begin{aligned} &\begin{bmatrix} OX_W^C & c^C & r^C & p_1^C & p_2^C & \vdots & p_n^C \end{bmatrix}_k^T \\ &\quad = \begin{bmatrix} OX_{W,k-1}^C & c_{measured}^C & r_{measured}^C & p_{1,k-1}^C & p_{2,k-1}^C & \vdots & p_{n,k-1}^C \end{bmatrix}_{k-1}^T \end{aligned} \tag{4}$$

Where

$$c_{measured}^C = v_{measured}^C \Delta t + \frac{1}{2} a_{measured}^C \Delta t^2$$

$$r_{measured}^C = r_{k-1}^C + w_{measured}^C$$

1.2.3 Measurement Model

Each observed feature is related to the camera motion through the measurement model (). This relationship enables a correction on the camera motion and features parameters based on the features' locations observed in the image.

For a feature p_i^C, the vector h^Rpointing from the predicted camera location to the feature initialization position is

$$h_k^R = \begin{bmatrix} x_i^C \\ y_i^C \\ z_i^C \end{bmatrix}_k - \begin{bmatrix} c_x^C \\ c_y^C \\ c_z^C \end{bmatrix}_k \tag{5}$$

The normalized vector pointing from the predicted camera position to the feature at time k is then

$$h_k^C = Q^{-1}\left(r_k^C\right)\left(\rho_k h_k^R + m\left(\varphi_k^C, \theta_k^C\right)\right) \tag{6}$$

where $Q^{-1}(r_k^C)$ is the inverse rotation matrix from the camera frame at time $k-1$ to camera frame at time k. From vector h_k^C, the feature location on image plane can be found by

$$h_k^U = \begin{bmatrix} u_k \\ v_k \end{bmatrix} = \begin{bmatrix} \frac{s_x h_{y,k}^C}{h_{x,k}^C} \\ \frac{s_y h_{z,k}^C}{h_{x,k}^C} \end{bmatrix} \tag{7}$$

where s_x and s_y is the scaling factor of the projection, obtained through camera calibration.

1.2.4 Composition Step

Update step corrects the camera motion and feature location in camera frame k-1. To continue to the next cycle of tracking, all parameter must be transform to camera frame k. World reference point coordinate and orientation from k-1 to k is related by

$$\begin{bmatrix} O_x^{C_k} \\ \\ O_y^{C} \\ \\ O_z^{C} \end{bmatrix}_k = R^{-1}(r_k^{C_{k-1}}) \left(\begin{bmatrix} O_x^{C_{k-1}} \\ \\ O_y^{C_{k-1}} \\ \\ O_z^{C_{k-1}} \end{bmatrix}_k - \begin{bmatrix} c_x^{C_{k-1}} \\ \\ c_y^{C_{k-1}} \\ \\ c_z^{C_{k-1}} \end{bmatrix}_k \right) \tag{8}$$

$$\begin{bmatrix} W_x^{C_k} \\ \\ W_y^{C_k} \\ \\ W_z^{C_k} \end{bmatrix}_{k-1} = \begin{bmatrix} W_x^{C_{k-1}} \\ \\ W_y^{C_{k-1}} \\ \\ W_z^{C_{k-1}} \end{bmatrix}_k - r^{C_{k-1}} \tag{9}$$

Feature parameters in new camera frame are related to the previous frame by

$$\begin{bmatrix} x_i^{C_k} \\ \\ y_i^{C_k} \\ \\ z_i^{C_k} \end{bmatrix}_k = Q^{-1}(r^{C_{k-1}}) \left(\begin{bmatrix} x_i^{C_{k-1}} \\ \\ y_i^{C_{k-1}} \\ \\ z_i^{C_{k-1}} \end{bmatrix}_k - \begin{bmatrix} c_x^{C_{k-1}} \\ \\ c_y^{C_{k-1}} \\ \\ c_z^{C_{k-1}} \end{bmatrix}_k \right) \tag{10}$$

$$\begin{bmatrix} \rho_i \\ \\ \varphi_i^{C_k} \\ \\ \theta_i^{C_k} \end{bmatrix}_k = \begin{bmatrix} \rho_{i,k} \\ \\ m^{-1}(R^{-1}(r^{C_{k-1}})m(\varphi_{i,k}^{C_{k-1}}, \theta_{i,k}^{C_{k-1}}) \end{bmatrix} \tag{11}$$

where $m(\varphi_{i,k}^{C_{k-1}}, \theta_{i,k}^{C_{k-1}})$ is the unit vector pointing from the initialization point to the feature seen by the camera at step $k-1$

The covariance matrix is also affected by this transform. Therefore must be updated. The new covariance matrix is related to the old one by

$$P_k^{C_k} = J_{C_{k-1} \to C_k} P_k^{C_{k-1}} J_{C_{k-1} \to C_k}^T \tag{12}$$

The calculation of $J_{C_{k-1} \to C_k}$ is the same as the linearization of prediction matrix in section Method 2.

In order to apply the correction and update the camera reference frame to the new camera position, an additional composition step is necessary. The world reference frame parameters and features parameters are updated by applying reference frame transformation from the camera location at time $k-1$ to camera location at time k. The EKF covariance matrix P_k is also updated through

$$P_k = J P_k J^T \tag{13}$$

where J is the Jacobian of the composition equations.

1.3 Initialization

1.3.1 Initialize the State Vector

State vectors are initialized at the first frame. The world origin coordinate and orientation, camera motions, and the feature reference points are all initialized to zeros, with variance equals to the smallest machine number. Thus,

$$OX_W^C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{14}$$

$$c^C = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{15}$$

$$r^C = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{16}$$

$$p_i^C = \begin{bmatrix} 0 & 0 & 0 & \rho_i & \varphi_i & \theta_i \end{bmatrix} \tag{17}$$

The inverse distance ρ of all features are initialized to 0.1 because we are dealing with long distance object. The features elevation-azimuth pair $[\varphi_i^C, \theta_i^C]$ is extracted from features coordinates in image plane. First, a vector pointing from camera optical center to feature can be defined by

$$h^C = \begin{bmatrix} h_x^C \\ h_y^C \\ h_z^C \end{bmatrix} = \begin{bmatrix} 1 \\ u \cdot s_x \\ v \cdot s_y \end{bmatrix} \tag{18}$$

Where $[uv]$ is the feature coordinate in the image, and $[s_x s_y]$is the scaling factor of the projection from the scene to image plane. The elevation-azimuth pair $[\varphi_i^C, \theta_i^C]$ can then be directly calculated from h^C

$$\varphi = arctan\left(\frac{h_z^C}{\sqrt{h_x^{C2} + h_y^{C2}}}\right) \tag{19}$$

$$\theta = arctan\left(\frac{h_y^C}{h_x^C}\right) \tag{20}$$

1.3.2 Initialized the State Covariance Matrix

Because the world origin is defined at the first frame, it enables initializing the filter with minimum variance, which helps reducing the lower bound of the filter error. The covariance matrix of the world coordinate and orientation, and the camera motion is

$$P = I_{12\times 12} \cdot \epsilon \tag{21}$$

where I is a 12×12 identity matrix, and ϵ is the lowest significant bit (LSB) of a machine.

The covariance of features is added one by one as there is correlation between them. For every new feature added, the new covariance matrix becomes

$$P_{new} = J \begin{bmatrix} P_{old} & 0 \\ 0 & R \end{bmatrix} J^T \tag{22}$$

where P_{old} isthe covariance matrix of the existing state vector, and the initial P_{old} is defined in . Matrix R is the covariance matrix of the variable in features initialization.

$$R = \begin{bmatrix} \sigma_{x_i^C} & & & & & \\ & \sigma_{y_i^C} & & 0 & & \\ & & \sigma_{z_i^C} & & & \\ & & & \sigma_\rho & & \\ & 0 & & & \sigma_{image} & \\ & & & & & \sigma_{image} \end{bmatrix} = \begin{bmatrix} \epsilon & & & & & \\ & \epsilon & & 0 & & \\ & & \epsilon & & & \\ & & & 0.1 & & \\ & 0 & & & 1 & \\ & & & & & 1 \end{bmatrix} \tag{23}$$

where$[\sigma_{x_i^C} \sigma_{y_i^C} \sigma_{z_i^C}]$ is the uncertainty of the camera optical center position, initialized to ϵ.$\sigma_{image}$is the image plane pixel variance, set to 1. $\sigma_\rho$is the uncertainty of the inverse distance. Because the filter mainly deals with distance feature, σ_ρ is initialized to 0.1 to cover any distance from 50 meters to infinity.

J in equation?? is the Jacobian matrix for the initialization equation.

$$J = \begin{bmatrix} & & & & & & 0 \\ & & & I & & & \vdots \\ & & & & & & 0 \\ \frac{\partial p_i}{\partial OX_W^C} & \frac{\partial p_i}{\partial c^C} & \frac{\partial p_i}{\partial r^C} & 0 & \dots & 0 & \frac{\partial p_i}{\partial g_i} \end{bmatrix} \tag{24}$$

Whenever a new feature is added, it's initial position is always $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ in the camera centric coordinate, and the $\begin{bmatrix} \rho & \varphi & \theta \end{bmatrix}$ parameters are not a function of OX_W^C, c^C,orr^C, therefore

$$\frac{\partial p_i}{\partial OX_W^C} = 0_{6\times 6} \tag{25}$$

$$\frac{\partial p_i}{\partial c^C} = 0_{6\times 3}\frac{\partial p_i}{\partial r^C} = 0_{6\times 3} \tag{26}$$

J can then be simplified as

$$J = \begin{bmatrix} I & 0 & \\ 0 & \frac{\partial p_i}{\partial g_i} & \end{bmatrix} \tag{27}$$

whereg_i includes the variable in matrix R: $g_i = [x_i^C \quad y_i^C \quad z_i^C \quad \rho_i \quad u_i \quad v_i]$. Then

$$
\frac{\partial p_i}{\partial g_i} = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & & \\ & \frac{\partial \rho_i}{\partial \rho_i} & \frac{\partial \rho_i}{\partial u_i} & \frac{\partial \rho_i}{\partial v_i} \\ 0_{3\times3} & \frac{\partial \varphi_i^C}{\partial \rho_i} & \frac{\partial \varphi_i^C}{\partial u_i} & \frac{\partial \varphi_i^C}{\partial v_i} \\ & \frac{\partial \theta_i^C}{\partial \rho_i} & \frac{\partial \theta_i^C}{\partial u_i} & \frac{\partial \theta_i^C}{\partial v_i} \end{bmatrix} = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & & \\ & 1 & 0 & 0 \\ 0_{3\times3} & 0 & \frac{\partial \varphi_i^C}{\partial u_i} & \frac{\partial \varphi_i^C}{\partial v_i} \\ & 0 & \frac{\partial \theta_i^C}{\partial u_i} & \frac{\partial \theta_i^C}{\partial v_i} \end{bmatrix} \tag{28}
$$

Based on the rule of derivation,

$$
\frac{\partial \varphi_i^C}{\partial u_i} = \frac{\partial \varphi_i^C}{\partial h^C} \frac{\partial h^C}{\partial u_i}
$$

$$
\frac{\partial \theta_i^C}{\partial u_i} = \frac{\partial \theta_i^C}{\partial h^C} \frac{\partial h^C}{\partial u_i}
$$

$$
\frac{\partial \varphi_i^C}{\partial v_i} = \frac{\partial \varphi_i^C}{\partial h^C} \frac{\partial h^C}{\partial v_i}
$$

$$
\frac{\partial \theta_i^C}{\partial v_i} = \frac{\partial \theta_i^C}{\partial h^C} \frac{\partial h^C}{\partial v_i}
$$

and

$$
\frac{\partial \varphi_i^C}{\partial h^C} = \begin{bmatrix} \frac{\partial \varphi_i^C}{\partial h_x^C} & \frac{\partial \varphi_i^C}{\partial h_y^C} & \frac{\partial \varphi_i^C}{\partial h_z^C} \end{bmatrix} = \begin{bmatrix} \frac{-h_x^C \cdot h_z^C}{({h_x^C}^2 + {h_y^C}^2 + {h_z^C}^2)\sqrt{{h_x^C}^2 + {h_y^C}^2}} \\ \frac{-h_y^W \cdot h_z^W}{({h_x^C}^2 + {h_y^C}^2 + {h_z^C}^2)\sqrt{{h_x^C}^2 + {h_y^C}^2}} \\ \frac{\sqrt{{h_x^C}^2 + {h_y^C}^2}}{({h_x^C}^2 + {h_y^C}^2 + {h_z^C}^2)} \end{bmatrix}^T \tag{29}
$$

$$
\frac{\partial \theta_i^C}{\partial h^C} = \begin{bmatrix} \frac{\partial \theta_i^C}{\partial h_x^C} & \frac{\partial \theta_i^C}{\partial h_y^C} & \frac{\partial \theta_i^C}{\partial h_z^C} \end{bmatrix} = \begin{bmatrix} -\frac{h_y^C}{{h_x^C}^2 + {h_y^C}^2} & \frac{h_x^C}{{h_x^C}^2 + {h_y^C}^2} & 0 \end{bmatrix} \tag{30}
$$

$$\frac{\partial h^C}{\partial u_i} = \begin{bmatrix} \frac{\partial h_x^C}{\partial u_i} \\ \frac{\partial h_y^C}{\partial u_i} \\ \frac{\partial h_z^C}{\partial u_i} \end{bmatrix} = \begin{bmatrix} 0 \\ s_x \\ 0 \end{bmatrix} \tag{31}$$

$$\frac{\partial h^C}{\partial v_i} = \begin{bmatrix} \frac{\partial h_x^C}{\partial v_i} \\ \frac{\partial h_y^C}{\partial v_i} \\ \frac{\partial h_z^C}{\partial v_i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ s_y \end{bmatrix} \tag{32}$$

Linearization of Measurement Model

The measurement model of the algorithm is not linear, and must be linearized by evaluating first derivative of the measurement equation at the predicted camera location. Since the measurement model is not a function of world origin coordinate and orientation, the Jacobian matrix of the measurement model is

$$\frac{\partial h_k^U(x)}{\partial x} = \begin{bmatrix} 0_{2n\times 6} & \frac{\partial h_k^U(x)}{\partial c^C} & \frac{\partial h_k^U(x)}{\partial r^C} & \frac{\partial h_k^U(x)}{\partial p_1} & \dots & \frac{\partial h_k^U(x)}{\partial p_n} \end{bmatrix} = \begin{bmatrix} 0_{2n\times 6} & \frac{\partial h_k^U}{\partial h_k^C}\cdot\frac{\partial h_k^C}{\partial c^C} & \frac{\partial h_k^U}{\partial h_k^C}\cdot\frac{\partial h_k^C}{\partial r^C} & \frac{\partial h_k^U}{\partial h_k^C}\cdot\frac{\partial h_k^C}{\partial p_1} & \dots & \frac{\partial h_k^U}{\partial h_k^C}\cdot\frac{\partial h_k^C}{\partial p_n} \end{bmatrix} \tag{33}$$

Elements in the matrix above are given in - . The formula for calculating Q^{-1}in - can be found in

$$\frac{\partial h_k^U}{\partial h_k^C} = \begin{bmatrix} -\frac{s_x h_{y,k}^C}{{h_{x,k}^C}^2} & \frac{s_x}{h_{x,k}^C} & 0 \\ -\frac{s_y h_{z,k}^C}{{h_{x,k}^C}^2} & 0 & \frac{s_y}{h_{x,k}^C} \end{bmatrix} \tag{34}$$

$$\frac{\partial h^C}{\partial c^C} = -R^{-1}(r_k^C)\rho_k \tag{35}$$

$$\frac{\partial h(x)}{\partial r^C} = \begin{bmatrix} \frac{\partial h^C}{\partial r_x^C} & \frac{\partial h^C}{\partial r_y^C} & \frac{\partial h^C}{\partial r_z^C} \end{bmatrix} \tag{36}$$

List of References

[1] R. E. Kalman. "A new approach to linear filtering and prediction problems." *Journal of Basic Engineering* **82**(1), 35–45. ISSN 0098-2202 (1960).

[2] H. Sorenson. "Least-squares estimation: from gauss to kalman." *IEEE Spectrum* **7**(7), 63–68. ISSN 0018-9235 (1970).

[3] Analytic Sciences Corporation. *Applied optimal estimation*. M.I.T. Press, Cambridge, Mass. ISBN 0262200279 (1974).

[4] M. S. Grewal. *Kalman filtering: theory and practice*. Prentice-Hall information and system sciences series. Prentice-Hall, Englewood Cliffs, N.J. ISBN 013211335X (1993).

[5] F. L. Lewis. *Optimal estimation: with an introduction to stochastic control theory*. Wiley, New York. ISBN 0471837415 (1986).

[6] R. G. Brown. *Introduction to random signals and applied Kalman filtering*. John Wiley & Sons, Inc., New York, 2nd ed edition. ISBN 0471525731 (1993).

[7] P. S. Maybeck. *Stochastic models, estimation and control*. Number v.141 in Mathematics in science and engineering. Academic Press, New York. ISBN 0124807011 (1979).

[8] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping: part i." *IEEE Robotics Automation Magazine* **13**(2), 99–110. ISSN 1070-9932 (2006).

[9] T. Bailey and H. Durrant-Whyte. "Simultaneous localization and mapping (SLAM): part II." *IEEE Robotics Automation Magazine* **13**(3), 108–117. ISSN 1070-9932 (2006).

[10] R. C. Smith and P. Cheeseman. "On the representation and estimation of spatial uncertainty." *The International Journal of Robotics Research* **5**(4), 56–68. ISSN 0278-3649, 1741-3176 (1986).

[11] H. Durrant-Whyte. “Uncertain geometry in robotics.” *IEEE Journal of Robotics and Automation* **4**(1), 23–31. ISSN 0882-4967 (1988).

[12] J. Leonard and H. Durrant-Whyte. “Simultaneous map building and localization for an autonomous mobile robot.” In “IEEE/RSJ International Workshop on Intelligent Robots and Systems ’91. ’Intelligence for Mechanical Systems, Proceedings IROS ’91,” pages 1442–1447 vol.3 (1991).

[13] H. Durrant-Whyte, D. Rye, and E. Nebot. “Localization of autonomous guided vehicles.” In G. Giralt and G. H. P. Dr.-Ing, editors, “Robotics Research,” pages 613–625. Springer London. ISBN 978-1-4471-1257-0, 978-1-4471-1021-7 (1996).

[14] M. Csorba, J. K. Uhlmann, and H. F. Durrant-Whyte. “New approach to simultaneous localization and dynamic map building.” volume 2738, pages 26–36 (1996).

[15] M. Csorba. *Simultaneous Localisation and Map Building* (1997).

[16] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. “A solution to the simultaneous localization and map building (SLAM) problem.” *IEEE Transactions on Robotics and Automation* **17**(3), 229–241. ISSN 1042-296X (2001).

[17] G. P. Huang, A. Mourikis, and S. Roumeliotis. “Analysis and improvement of the consistency of extended kalman filter based SLAM.” In “IEEE International Conference on Robotics and Automation, 2008. ICRA 2008,” pages 473–479 (2008).

[18] S. Huang and G. Dissanayake. “Convergence and consistency analysis for extended kalman filter based SLAM.” *IEEE Transactions on Robotics* **23**(5), 1036–1049. ISSN 1552-3098 (2007).