0.1 Vision Based Sensor Overview

The pinhole camera model, is an approximation of the CMOS machine vision camera that is actually being used. To account for the errors that such an approximation may yield, it is proposed that the *projected* coordinates (u, v) be warped with a *radial distortion* as is done in [?], in order to obtain a new *distortion* coordinate (u_d, v_d) that will better resemble the one which the camera will provide. This radial distortion is mathematically shown as follows

$$\begin{aligned} u_d - u_0 &= \frac{u - u_0}{(1 + k_1^2 r^2 + k_2 r^4)} \\ v_d - v_0 &= \frac{v - v_0}{(1 + k_1^2 r^2 + k_2 r^4)}, \\ r &= \sqrt{(u - u_0)^2 + (v - v_0)^2}. \end{aligned} \tag{1}$$

0.1.1 Camera Calibration

After using a 9 × 6 chessboard pattern for camera calibration in OpenCV [?], the calibration parameters of the camera used in this particular implementation are as follows: $fk_u = fk_v = 520$ pixels, principal point $(u_0, v_0) = (315, 239)$ and $K = 0.982$ for capture at a resolution of 680 × 480.

0.2 Update Step

0.2.1 Measurement Model

The correction step of the Extended Kalman filter aims to ultimately correct the previously estimated robot pose and landmark position through exterior sensor measurements. With regard to the implementation proposed in this paper, these measurements are obtained through the use of a camera. The measurement process generally involves a measurement estimate that incorporates uncertainty.

With reference to figure 0.2.1, a feature's cartesian position can be described through a cartesian vector $\mathbf{h}_i^W(\bar{\boldsymbol{\mu}})$, where the feature's cartesian **point** is shown in relation to the camera's centre:

$$\mathbf{h}_i^W(\bar{\boldsymbol{\mu}}) = \mathbf{R}^{CW}\left(\mathbf{y}_i^W - \mathbf{r}^W\right) = \left(\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} - \mathbf{r}^W\right) \tag{2}$$

the subscript i corresponds a directional vector $\mathbf{h}^C$ from it's cartesian position $\mathbf{r}^W$ to the cartesian position of a given landmark $\mathbf{y}^W$.

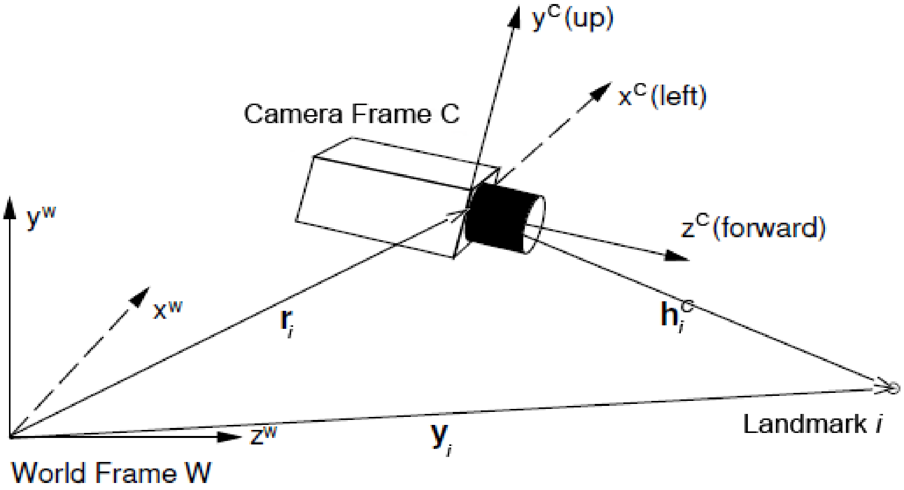

Figure 1: Davison's MonoSLAM representation of the reference frames

A camera however, cannot directly measure a cartesian vector. Instead, a camera measurement (based on the model presented) obtains a vector $\mathbf{h}_i$ that is a function of $\mathbf{h}_i^W$. This vector describes a given feature's horizontal and vertical image positions (u, v). For an undistorted image, the vector $\mathbf{h}_i$, more commonly referred to as the measurement function is defined as follows:

$$\mathbf{h}_i = \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} u_0 - fk_u \dfrac{h_{i,x}^R}{h_{i,z}^R} \\ v_0 - fk_v \dfrac{h_{i,y}^R}{h_{i,z}^R} \end{pmatrix} \tag{3}$$

where u_0 and v_0 represent the principal point and fk_u and fk_v are the camera calibration parameters described in Appendix **??**.
It is evident from the model presented in equation 3 cannot be directly inverted to obtain a feature's position. The projection of a feature onto the camera's image plane removes any information required to directly obtain the depth of the feature.

0.2.2 Feature Matching

The following section discusses the measurement of a feature *fully* initialised within the SLAM map. The measurement process seeks to initially estimate the cartesian position of a given feature $\mathbf{y}_i$ within the SLAM map. Thereafter, the feature can be compared via a matching sequence. Generally, feature matching is conducted using a normalised cross-correlation search, where a 2D template of the 3D feature is scanned is across the entire image (at each pixel location) until a peak is obtained. MonoSLAM however, seeks to utilise an *active* approach for matching, minimising the the search field and improving efficiency.

The EKF inherently contains information that may be utilised in order to prohibit a full cross-correlation search. The measurement function $\mathbf{h}(\bar{\mu})$ for instance, provides an estimate for a given features location, namely $\mathbf{u}_d = (u_d, v_d)$. Knowledge of this location therefore allows an active search region to be described within the vicinity of this location. The location estimate of the feature is not the only information regarding the feature that is available as a result of the EKF. Additionally, the uncertainty regarding a given feature's location is stored within the state vector covariance matrix $\mathbf{P}_{nN}$. This information can be used to determine the size of the active search region surrounding the location estimate; where the size of the search region is directly proportional to the uncertainty of it's location. If the feature cannot be matched within the aforementioned search region, it cannot contribute to the correction of the robot's pose estimate and is therefore deleted form the SLAM map. The aforementioned process of defining the active search region can be mathematically defined through the *innovation covariance matrix* $\mathbf{S}_i$:

$$\mathbf{S}_i = \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{x}_v} P_{xx} \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{x}_v}^T + \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{x}_v} P_{xy_i} \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{y}_i}^T + \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{y}_i} P_{y_i x} \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{x}_v}^T + \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{y}_i} P_{y_i y_i} \frac{\partial \mathbf{u}_{d,i}}{\partial \mathbf{y}_i}^T + \mathbf{R}_v \quad (4)$$

The symmetric 2×2 matrix $\mathbf{S}_i$ represents a 2D Gaussian PDF around the estimated image coordinate. The innovation covariance matrix can then be used to determine an active region the a given feature should lie within. Typically, the active search region is defined to confine within 3 standard deviations (3σ) of the mean.

Furthermore, the innovation matrix provides a measure of the amount of content expected within an eventual actual measurement $\mathbf{z}_i$. In the event that many potential measurements are available, features containing a higher $\mathbf{S}_i$ present the EKF with more information regarding the camera's position. Candidates for feature estimates are thus chosen according to those that present the most information regarding the position estimate. Feature searches per sampling instance are generally limited (usually about 12 features) due to computational constrains.

Finally, as described in [Davison], an active search will always reduce the area of the template matching search region at the potential *additional* cost of calculating the reduced search region.

0.2.3 Feature Initialisation

The inherent disadvantage of a monocular camera, as previously mentioned, is the inability to immediately provide an estimate for the depth of a feature. As a result, a given feature is required to be observed at various viewpoints before its depth can be approximated through a multiple view triangulation. Instead, Davison et al. presents an alternative approach whereby a feature is initialised to lie along an infinite 3D line. This line, originating from the position at which the camera is estimated, extends indefinitely in the direction of the feature. The depth of the feature lies somewhere along the aforementioned line. This depth can be modelled as a uniformly distributed set of discrete depth hypothesis. Briefly, the feature's depth can be interpreted as a 1D probability density, represented only by particle distribution instead. The feature is can then *partially* initialised in the SLAM map as follows:

$$\mathbf{y}_{pi} = \begin{pmatrix} \mathbf{r}_i^W \\ \hat{\mathbf{h}}_i^W \end{pmatrix} \tag{5}$$

where $\mathbf{r}_i^W$ represents the origin of the line and $\hat{\mathbf{h}}_i^W$ is a unit vector representing its direction. The uncertainty describing the aforementioned entities are Gaussian in nature.

After a feature has been partially initialised, it can be assumed that the feature is re-observed and that each additional observation improves the depth estimate. The particle filter based depth estimation process itself is to a large extent complex, and is explained in more detain in . Intuitively, the depth estimation process can be explained as follows: each particle in the particle set is projected into the image and subsequently matched across each observation. The resulting observations transform the initially uniformly distributed depth probability into one that better resembles a Gaussian density. Once the depth covariance is below a certain threshold, the depth is approximated with a Guassian probability density. Thereafter a feature becomes *fully* initialised, assigned with a standard 3D Gaussian representation.

0.2.4 Map Management

Map management forms an integral role in the realisation of the MonoSLAM algorithm. A real-time algorithm, as proposed in this paper, relies on efficient and accurate decisions regarding features within the SLAM map. As a result, a strict protocol is followed in in order to realise a successful real-time algorithm.