

Chapter 9

UKF-Based Image Filtering and 3D Reconstruction



Abdulkader Joukhadar, Dalia Kass Hanna, and Etezaz Abo Al-Izam

Acronyms

2D, 3D	Two- and three-dimensional space
4WDDMR	Four-wheeled differential drive mobile robot
AO	Absolute orientation
BA	Bundle adjustment
Bel	Degree of belief
EKF	Extended Kalman filter
EO	Exterior orientation
GRV	Gaussian random variable
IO	Interior orientation
KF	Standard Kalman filter
MAP	Maximize the posterior estimation
MLE	Maximum likelihood estimator
pdf	Probability density function
RANSAC	Random sample consensus
RO	Relative orientation
SFM	Structure from motion problem
SIFT	Scale invariant feature transform
SURF	Speeded up robust features
UKF	Unscented Kalman filter
VSLAM	Visual simultaneous localization and mapping

A. Joukhadar (✉) · D. Kass Hanna
Department of Mechatronics Engineering, Faculty of Electrical and Electronic Engineering,
University of Aleppo, Aleppo, Syria
e-mail: ajoukhadar@alepuniv.edu.sy; daliakasshanna@alepuniv.edu.sy

E. Abo Al-Izam
Department of Computer Engineering, Faculty of Electrical and Electronic Engineering,
University of Aleppo, Aleppo, Syria

9.1 Introduction

Recently, techniques for solving the problem of reconstructing a 3D model of the environment have drawn significant attention from both the robotics vision communities and computer vision. Many variants of these techniques have started to make an impact in a wide range of robotics applications such as robot navigation, obstacle avoidance, and visual simultaneous localization and mapping (VSLAM) based on stereo vision [1]. Structure from motion problem (SFM) is defined as a technique for 3D reconstruction from a set of 2D images and some geometrical constraints [2].

SFM methods perform a *bundle adjustment (BA)* optimization of the total geometry in order to obtain an accurate 3D model of the scene. However, this is computationally very expensive and cannot be implemented in a real-time application [3]. On the other hand, *simultaneous localization and mapping (SLAM)* is a problem faced recently in robotics community, essentially addressing the hard real-time mapping and navigation problem by “sequential” interactive local estimation of the structure and motion [4].

The SLAM problem tries to answer the following central question: “*Is it possible for an autonomous robot starting at an unknown location in an unknown environment to build a map of the environment while simultaneously using this map to compute the vehicle’s location?*” Solving this problem allows to develop a truly autonomous robot and navigate safely around the environment [5].

The stochastic nature of the mobile robot motion with noisy measurement data complicates the coupling between navigation and mapping that is inherent SLAM [6]. Many successful SLAM algorithms address these issues by formulating the problem in a probabilistic manner, tracking the joint posterior over the robot pose and map.

Probabilistic Bayesian filter is the strategy to incorporate uncertainty for all possible robot poses with a probability density function (pdf), which is the degree of belief (Bel) of the robot moves [7].

The estimation problem can be solved by KF-based approaches or particle filter (i.e., online VSLAM) for real-time application or bundle adjustment (i.e., standard SFM) for an offline application [7]. In the medical field, stereo vision obtained a 3D vision, which improved the accuracy of surgery and reduced the time required for surgery and errors that may occur. A novel SLAM algorithm proposed by [8] aimed at advancing the state-of-art in image-guided surgery using stochastic models and KF framework to recursively estimate the configuration of the high degree of freedom snake surgical robot using stereo vision. VSLAM estimates the camera pose by implementing *epipolar geometry* on the static feature correspondences as shown in Fig. 9.1. The dynamic features are regarded as outliers and excluded from the computation.

From Fig. 9.1, the static corner has been extracted as a correspondence point in order to estimate the camera orientation. The computer vision community has developed a large number of feature extraction techniques (e.g., Harris corner



Fig. 9.1 Feature correspondence extraction

detector, scale invariant feature transform (SIFT), and speeded-up robust features (SURF)). Unfortunately, these feature-matching techniques do not guarantee perfect correspondences, especially when the data contains outliers [3]. Implementation of robust estimators RANSAC (random sample consensus) is useful to reject outliers and handle false correspondences. On the other hand, deep learning techniques can process the image sequences directly to compute the correspondences in real time [3].

This chapter is structured as follows: Sect. 9.1 introduces the basic estimation techniques; Sect. 9.2 presents the brief of stereo vision, camera calibration, projection, and epipolar function; Sect. 9.3 exhibits the uncertainties and error source in vision system; finally, Sect. 9.4 shows examples of VSLAM based on UKF algorithm and stereo vision.

9.2 Kalman Filter Framework-Based Probabilistic Inference

The following problem commonly recurs in Computer Vision and Autonomous Robotics algorithms: estimate the values of unknown parameters (robot posture, camera orientation, etc.), given a number of measurements (sensory data, images, feature points, etc.). These kinds of problems are called *inverse problem* because they involve in estimating unknown model parameters instead of simulating the forward formation equations [9]. However, a model of uncertainty sources needs to be introduced in order to have a reasonable algorithm. Such inference problems from noisy data are called *probabilistic inference* [10]. In this section, the measurement update equation for the KF, EKF, and UKF is derived starting from the maximum

likelihood (the joint probability density function (pdf)), work through Bayesian rule, Gauss–Newton iterated nonlinear least squares method, and practical nonlinear Bayes filter (i.e., standard Kalman filter (SKF), extended Kalman filter (EKF)-based Taylor linearization, and unscented Kalman filter (UKF)-based stochastic linearization).

9.2.1 Maximum Likelihood Estimator (MLE)

The general form of noisy measurement model is given by (9.1):

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (9.1)$$

where \mathbf{z}_k is the noisy measurement vector, \mathbf{x}_k is the unknown state vector, $h(\cdot)$ is the associated nonlinear measurement model, which maps the unknown into that particular measurement, and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$ is a normal *Gaussian random variable* (GRV) with zero mean and covariance matrix \mathbf{R} . Given all of the noisy measurements $\mathbf{z} = \{\mathbf{z}_k\}$, the likelihood of having the observed $\{\mathbf{z}_k\}$ given a particular value of \mathbf{x} is given by (9.2):

$$L = p(\mathbf{z}|\mathbf{x}) = \prod_k p(\mathbf{z}_k|\mathbf{x}_k) = \prod_k p(\mathbf{z}_k|h(\mathbf{x}_k)) = \prod_k p(\mathbf{v}_k) \quad (9.2)$$

where $p(\mathbf{z}|\mathbf{x})$ is the joint probability distribution of the measurements \mathbf{z} with the unknown vector \mathbf{x} . To solve the inverse problem (if the distribution is unimodal Gaussian), the optimal estimate value for the unknown state vector \mathbf{x} in the absence of any prior model is that maximizes the likelihood function. if the distribution is unimodal Gaussian. However, if the probability is multimodal, it has several local maxima in likelihood, much more care is required [11].

The likelihood function can be written for the Gaussian noise as (9.3):

$$\begin{aligned} L &= \prod_k |2\pi\mathbf{R}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k))\right) \\ &= \prod_k |2\pi\mathbf{R}|^{-1/2} \exp\left(-\frac{1}{2}\|\mathbf{z}_k - \bar{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2\right) \end{aligned} \quad (9.3)$$

The norm $\|\mathbf{z}_k - \bar{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2$ is called the *Mahalanobis distance* [9]. It is used to measure the distance between the measurement \mathbf{z}_k with step time \mathbf{k} and the mean of Gaussian distribution $\bar{\mathbf{z}}_k$. Usually, it is more convenient to work with the negative log-likelihood, as a cost function [12] (9.4):

$$\begin{aligned} E = -\log L &= \frac{1}{2} \sum_k (\mathbf{z}_k - \bar{\mathbf{z}}_k)^T \mathbf{R}^{-1} (\mathbf{z}_k - \bar{\mathbf{z}}_k) + \log |2\pi\mathbf{R}| \\ &= \frac{1}{2} \sum_k \|\mathbf{z}_k - \bar{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2 + K \end{aligned} \quad (9.4)$$

where $K = \sum_k \log |2\pi \mathbf{R}|$ is a constant independent of \mathbf{x} and can be dropped. The inverse covariance matrix \mathbf{R}^{-1} weights each measurement error residuals (i.e., the difference between the real measurement \mathbf{z}_k and the predicted mean value $\hat{\mathbf{z}}_k = \bar{\mathbf{z}}_k$).

Another form of negative log-likelihood can be written as (9.5):

$$E = -\log L = \sum_k \|\mathbf{z}_k - \hat{\mathbf{z}}_k\|_{\mathbf{R}^{-1}} \quad (9.5)$$

Consider the measurement noise is Gaussian and the measurement equation is linear (9.6):

$$\mathbf{z}_k = h(\mathbf{x}_k) = \mathbf{H}\mathbf{x}_k \quad (9.6)$$

where \mathbf{H} is the measurement matrix; in this case, the maximum likelihood estimate is given by the minimization of the quadratic function (9.7), which is a simple quadratic form in \mathbf{x}_k solved using linear least square algorithm:

$$E = \sum_k \|\mathbf{z}_k - h(\mathbf{x}_k)\|_{\mathbf{R}^{-1}} = \sum_k (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k)^T \mathbf{R}^{-1} (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k) \quad (9.7)$$

9.2.2 Probabilistic Inference and Bayesian Rule

In some cases, the range of possible solution consistent with the measurements is too large to be useful, and any progress cannot be made [11]. For example, MLE estimates each pixel separately based on just its noisy version to solve the problem of image filtering [13]. The difference between the Bayesian inference and the MLE method is that the starting point of Bayesian inference is to formally consider the unknown vector \mathbf{x}_k as a random vector with a prior distribution $p(\mathbf{x}_k)$, which is called the degree of belief (Bel), then the posterior distribution of \mathbf{x}_k can be computed by multiplying the measurements likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ by the prior Bel [14].

Consider the noisy measurement model given by (9.8):

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (9.8)$$

where $\mathbf{x}_k \sim N(\bar{\mathbf{x}}, \mathbf{P})$ is the unknown Gaussian random state vector with mean $\bar{\mathbf{x}}$ and state covariance matrix \mathbf{P} . The form of Bayesian rule is given by (9.9):

$$p(\mathbf{x}_k|\mathbf{z}_k) = \eta p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k) \quad (9.9)$$

where η is the normalizing constant [10]. The problem is to find \mathbf{x}_k that maximizes the posterior estimation (MAP).

Assume that the distributions of \mathbf{x}_k and \mathbf{z}_k are Gaussian (9.10):

$$\begin{aligned} p(\mathbf{z}_k | \mathbf{x}_k) &= |2\pi \mathbf{R}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k))\right) \\ p(\mathbf{x}_k) &= |2\pi \mathbf{P}_k|^{-1/2} \exp\left(-\frac{1}{2}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T \mathbf{P}_k^{-1} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)\right) \end{aligned} \quad (9.10)$$

The solution that maximizes $p(\mathbf{x}_k | \mathbf{z}_k)$ is the most probable value of the random vector and is equivalent to minimize its negative log, which reduces to the quadratic form [11] (9.11):

$$L = \frac{1}{2} \left((\mathbf{z}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k)) + (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T \mathbf{P}_k^{-1} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-) \right) \quad (9.11)$$

An algebraic equivalent way to maximize the posterior likelihood is to consider the prior estimate as a pseudo-observation and write a new observation vector [12] (9.12):

$$\mathbf{Z}_k = \begin{bmatrix} \mathbf{z}_k \\ \hat{\mathbf{x}}_k^- \end{bmatrix}, g(\mathbf{x}_k) = \begin{bmatrix} h(\mathbf{x}_k) \\ \hat{\mathbf{x}}_k^- \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_k \end{bmatrix} \quad (9.12)$$

which gives (9.13):

$$L = \frac{1}{2} (\mathbf{Z}_k - g(\mathbf{x}_k))^T \mathbf{C}^{-1} (\mathbf{Z}_k - g(\mathbf{x}_k)) \quad (9.13)$$

This is a nonlinear least squares problem of the form (9.14):

$$E = -\log L = \sum_k \|\mathbf{z}_k - \hat{\mathbf{z}}_k\|_{\mathbf{R}^{-1}}^2 \quad (9.14)$$

A useful approximation for small residual problems is the online stochastic Gauss-Newton method, which defines the sequence of iterates as [12] (9.15):

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- - \left(\mathbf{G}_k^T \mathbf{C}^{-1} \mathbf{G}_k \right)^{-1} \mathbf{G}_k^T \mathbf{C}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k)) \quad (9.15)$$

where $\mathbf{G}_k = \left. \frac{\partial g(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}=\bar{\mathbf{x}}}$ the Jacobian of $g(\mathbf{x}_k)$ with respect to the state vector \mathbf{x}_k .

The Gauss-Newton method is simply using the *matrix inversion lemma* [11] (9.16):

$$\begin{aligned} \left(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{P}_k^{-1} \right)^{-1} \mathbf{H}^T \mathbf{R}^{-1} &= \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \\ \left(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{P}_k^{-1} \right)^{-1} &= \mathbf{P}_k - \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \mathbf{H} \mathbf{P}_k \end{aligned} \quad (9.16)$$

Using (9.16), the Kalman equation is given by (9.17):

$$\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k}}^- - \mathbf{K}(\mathbf{z}_{\mathbf{k}} - h(\mathbf{x}_{\mathbf{k}})) \quad (9.17)$$

where $\mathbf{K}_{\mathbf{k}}$ is the Kalman gain (9.18):

$$\mathbf{K}_{\mathbf{k}} = \mathbf{P}_{\mathbf{k}}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{\mathbf{k}}\mathbf{H}^T + \mathbf{R})^{-1} \quad (9.18)$$

The update covariance matrix is approximated using (9.19):

$$\mathbf{P}_{\mathbf{k}+1} = (\mathbf{G}_{\mathbf{k}}^T\mathbf{C}^{-1}\mathbf{G}_{\mathbf{k}})^{-1} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_{\mathbf{k}}^{-1})^{-1} \quad (9.19)$$

where the Hessian $\mathbf{H}_{\mathbf{k}} = \left. \frac{\partial h(\mathbf{x}_{\mathbf{k}})}{\partial \mathbf{x}_{\mathbf{k}}} \right|_{\mathbf{x}=\bar{\mathbf{x}}}$ is the Jacobian of $h(\mathbf{x}_{\mathbf{k}})$ with respect to the state vector $\mathbf{x}_{\mathbf{k}}$.

The posterior covariance matrix using *matrix inversion lemma* (9.16) is given by (9.20):

$$\mathbf{P}_{\mathbf{k}+1} = (\mathbf{I} - \mathbf{K}_{\mathbf{k}}\mathbf{H})\mathbf{P}_{\mathbf{k}} \quad (9.20)$$

Many applications require an estimate for the uncertainty in this estimate such as KF, which require the computation of this uncertainty as posterior covariance matrix in order to optimally integrate new measurements with previously computed estimates [9].

9.2.3 Bayes Filter and Belief Update

In this section, the formulation of optimal recursive discrete time Bayesian filters (e.g., KF, EKF, and UKF) is presented as a practical estimator in terms of Bayes filter.

The basic elements of Bayesian filter are the initial belief $\text{Bel}(\mathbf{x}_{\mathbf{k}-1})$ containing preliminary information on the unknown vector $\mathbf{x}_{\mathbf{k}-1}$, the motion model $p(\mathbf{x}_{\mathbf{k}}|\mathbf{x}_{\mathbf{k}-1}, \mathbf{u}_{\mathbf{k}})$ as a probabilistic model of the discrete time state space, and the measurement model $p(\mathbf{z}_{\mathbf{k}}|\mathbf{x}_{\mathbf{k}})$ determining the stochastic mapping from the state vector to the measurement, where

- $\mathbf{x}_{\mathbf{k}} \in \mathfrak{N}^n$ is the unknown state space vector on time step \mathbf{k} .
- $\mathbf{u}_{\mathbf{k}} \in \mathfrak{N}^L$ is the control vector on time step \mathbf{k} .
- $\mathbf{z}_{\mathbf{k}} \in \mathfrak{N}^m$ is the observation vector on time step \mathbf{k} .

Using Markov assumption, these vectors are conditionally independent of past values. Bayes filter applies two rules successively to predict the system state [10] (9.21):

$$\begin{aligned}
\text{Prediction : } \overline{\text{Bel}}(\mathbf{x}_k) &= \sum_k p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \text{Bel}(\mathbf{x}_{k-1}) \\
\text{correction : } \text{Bel}(\mathbf{x}_k) &= \eta p(\mathbf{z}_k | \mathbf{x}_k) \overline{\text{Bel}}(\mathbf{x}_k)
\end{aligned} \tag{9.21}$$

The prior predictive $\overline{\text{Bel}}(\mathbf{x}_k)$ is calculated just before the measurement \mathbf{z}_k using the control vector \mathbf{u}_k ; this step is called prediction (or *action phase*). Next, the state estimate belief given in the action phase is corrected using sensor measurement; this step is called correction (or *perception phase*).

9.2.3.1 KF Framework

In many vision applications, the object is tracked from frame to frame as it moves. *Kalman Filter* (1960) has been regarded as the optimal solution to many visual motion tracking and data prediction tasks [13].

The standard KF derivation is given here in the practical use of probabilistic inference [15]. Consider a noisy linear system given by (9.22):

$$\begin{aligned}
\mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}_1\mathbf{u}_k + \mathbf{B}_2\boldsymbol{\omega}_k \\
\mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{v}_k
\end{aligned} \tag{9.22}$$

where $\mathbf{x}_k, \mathbf{x}_{k-1}$ are the current and previous state vector, $\mathbf{A}_{n \times n}$ is the linear state transition matrix of the dynamic model, $\mathbf{B}_{1n \times L}$ is the control matrix, $\mathbf{B}_{2n \times L}$ is the input noise matrix, $\mathbf{H}_{n \times m}$ is the measurement model matrix, $\boldsymbol{\omega}_k \sim N(\mathbf{0}, \mathbf{Q})$ is an additive Gaussian state noise, and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$ is the Gaussian measurement noise. The KF equations can be derived as follows [16]:

1. **Prediction phase:** The motion model causes a drift in the previous estimate, while the additive noise increases the system disbelief.

First, apply the motion model and compute the joint distribution of the Gaussian state \mathbf{x}_k given the initial state $\mathbf{x}_{k-1} \sim N(\bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})$ and the input \mathbf{u}_k by (9.23):

$$\hat{\mathbf{x}}_k^- \sim N\left(\mathbf{A}\bar{\mathbf{x}}_{k-1} + \mathbf{B}\bar{\mathbf{u}}_k, \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}\right) \tag{9.23}$$

Then, apply the measurement model and compute the joint distribution of the measurement \mathbf{z}_k given the predicted state $\hat{\mathbf{x}}_k^-$ by (9.24):

$$\hat{\mathbf{z}}_k \sim N\left(\mathbf{H}\hat{\mathbf{x}}_k^-, \mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}\right) \tag{9.24}$$

2. **Correction phase:** New measurements from the current frame introduce additional information that updates the prior estimate $\hat{\mathbf{x}}_k^-$ and restores some of the belief, by computing Kalman gain \mathbf{K}_k and updating the covariance matrix (9.25):

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}^T [\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}]^{-1} \\
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k) \\
\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-
\end{aligned} \tag{9.25}$$

9.2.3.2 EKF Linearization Technique

For nonlinear problems, such as SLAM, an EKF, which linearizes the motion and measurement model around the current estimate, is used. The important drawback of the EKF approach is that it uses the Taylor linearization dynamic model [17]. However, if the robot drives along a straight path, the distribution of mobile model in a plane has been observed by “Banana Shape distribution.” As uncertainty increases, the algorithm becomes inconsistent due to the normality assumption breaking down [18]. Consider a noisy nonlinear system given by (9.26):

$$\begin{aligned}
\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_k, \boldsymbol{\omega}_k) \\
\mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k
\end{aligned} \tag{9.26}$$

where $f(\cdot)$ is the nonlinear motion model, noisy by non-additive Gaussian noise $\boldsymbol{\omega}_k \sim N(\mathbf{0}, \mathbf{Q})$ and $h(\cdot)$ is the nonlinear measurement model, noisy by $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$.

Table 9.1 illustrates the pseudo code for EKF algorithm as Bayes filter.

9.2.3.3 UKF Stochastic Linearization Technique

The UKF is a Gaussian recursive Bayesian filtering algorithm to solve the probabilistic inference practically. It propagates and updates the system state using a set of deterministically chosen points called sigma points [11]. These points capture the mean and covariance of the state distribution. Filter each point using unscented transform through the nonlinear motion and measurement models [14], and determine the posterior state mean and state covariance to the third order of the nonlinear system. This is a form of statistical local linearization, which produces more accurate estimates than the analytic local linearization employed by the EKF [19]. The UKF algorithm includes three main stages. Table 9.2 shows the pseudo code for the UKF algorithm [16].

9.3 Stereo Vision System

Stereo vision is considered one of the most important recent applications [21]. It is still developing especially in the robotics vision application [22]. Stereo vision is used to form a 3D map of the robot environment, and landmarks are used in this map for localization and exploration [21].

Table 9.1 Extended Kalman filter ($\mathbf{x}_k - 1, \mathbf{P}_k - 1, \mathbf{u}_k, \mathbf{z}_k$)**Initialization:**

Initialize the prior knowledge $\mathbf{x}_k - 1, \mathbf{P}_k - 1, \mathbf{Q}, \mathbf{R}$:

$$\mathbf{x}_{k-1} = [\mathbf{x}_0]^T$$

$$\mathbf{P}_{k-1} = \text{diag}(\mathbf{P}_0) \in \mathfrak{N}^n, \mathbf{Q} = \text{diag}(\mathbf{Q}) \in \mathfrak{N}^L, \mathbf{R} = \text{diag}(\mathbf{R}) \in \mathfrak{N}^m$$

For each sample time k , do:

Prediction:

1. Apply the motion model and compute the mean and covariance:

$$\hat{\mathbf{x}}_k^- = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k); \quad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q})$$

$$\mathbf{P}_k^- = \mathbf{J}_x \mathbf{P}_{k-1} \mathbf{J}_x^T + \mathbf{J}_u \mathbf{Q} \mathbf{J}_u^T$$

where the Jacobian matrices $\mathbf{J}_x, \mathbf{J}_u$ are obtained by differentiating $f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$ with respect to \mathbf{x}_{k-1} and \mathbf{u}_k , respectively:

$$\mathbf{J}_x = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{x}_{k-1}} \right|_{\mathbf{x}=\hat{\mathbf{x}}}, \mathbf{J}_u = \left. \frac{\partial f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)}{\partial \mathbf{u}_k} \right|_{\mathbf{u}=\hat{\mathbf{u}}}$$

2. Apply the measurement model:

$$\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_k^-, \mathbf{v}_k); \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$$

Correction:

The EKF gain \mathbf{K}_k which minimizes the errors and updates the posterior mean and covariance ($\hat{\mathbf{x}}_k, \mathbf{P}_k$) is given by:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T \left[\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R} \right]^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-$$

return ($\hat{\mathbf{x}}_k, \mathbf{P}_k$)

Compute the Degree of Belief (Bel):

The determinant \mathbf{P}_k provides a good measure of uncertainty [10].

$$\text{Bel}_k = 1 - \|\mathbf{P}_k\|_2^{0.5}; \quad 0 \leq \text{Bel} \leq 1$$

end function

Figure 9.2 explains a low- and high-level image processing stages. In the low-level image processing stage, a camera calibration with distortion removal is carried out. Camera calibration includes the determination of the camera's intrinsic and extrinsic parameters. Accurate estimates of this geometry are necessary in order to relate image information to an external world coordinate system. On the other hand, in the high-level image processing stage, certain correspondence points are determined based on advanced algorithms for feature extraction [23]. These points are hence used to calculate the relative orientation (RO) as well as the absolute orientation (AO) using a set of control points whose relative coordinates and corresponding image points are known. Stereo vision-based system with SLAM algorithm is used to enable the robot to percept the environment around and within the robot playing area [24]. The computation of relative camera pose can be done using 7-point correspondences for an uncalibrated camera or 5-point for a calibrated camera from two views under by enforcing *epipolar geometry*. If the image correspondences are known, the relative pose between two images can be recovered up to a scale factor [13]. When the camera pose is recovered, one can easily reconstruct 3D points of the scene by intersecting two projection ray lines through triangulation [25]. As the rays do not

Table 9.2 Unscented Kalman filter (χ_{k-1} , \mathbf{P}_{k-1} , \mathbf{u}_k , \mathbf{z}_k)**Initialization:**

Initialize the state space vector χ_0 and covariance matrix \mathbf{P}_{k-1} , \mathbf{Q} , \mathbf{R} :

$$\chi_0 = [\mathbf{x}_0 \ \mathbf{0} \ \mathbf{0}]^T$$

$$\mathbf{P}_{k-1} = \text{diag}(\mathbf{P}_0, \mathbf{Q}, \mathbf{R})$$

For each sample time \mathbf{k} do:

Prediction:

1. Compute $2n - 1$ sigma points:

$$\chi_{k-1} = \left[\mathbf{x}_{k-1} \ \mathbf{x}_{k-1} + \gamma \sqrt{\mathbf{P}_{k-1}} \ \mathbf{x}_{k-1} - \gamma \sqrt{\mathbf{P}_{k-1}} \right]^T$$

where γ is a scalar parameter that determines how far the sigma points are dispersed away from the mean, and $\sqrt{\mathbf{P}_{k-1}}$ is computed using *Cholesky decomposition* [20].

2. Apply the motion model:

$$\chi_{\mathbf{k}} = f(\chi_{k-1}, \mathbf{u}_k, \mathbf{w}_k); \quad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q})$$

3. Compute the predicted sigma points mean $\hat{\chi}_{\mathbf{k}}^-$ and covariance $\mathbf{P}_{\chi\chi}^-$:

$$\hat{\chi}_{\mathbf{k}}^- = \sum_{i=0}^{2n} \omega_m^{[i]} \chi_{\mathbf{k}}^{[i]}$$

$$\mathbf{P}_{\chi\chi}^- = \sum_{i=0}^{2n} \omega_c^{[i]} \left(\chi_{\mathbf{k}}^{[i]} - \hat{\chi}_{\mathbf{k}}^- \right) \left(\chi_{\mathbf{k}}^{[i]} - \hat{\chi}_{\mathbf{k}}^- \right)^T$$

where $\omega_m^{[i]}$, $\omega_c^{[i]}$ are defined by the algorithm.

4. Propagate the new sigma points through the measurement model:

$$\mathbf{z}_{\mathbf{k}} = h \left(\chi_{\mathbf{k}}^{[i]}, \mathbf{v}_k \right); \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R})$$

5. Compute the new sigma points mean $\hat{\mathbf{z}}_{\mathbf{k}}$, the predicted measurement covariance \mathbf{P}_{zz} , and the state and measurement cross-covariance $\mathbf{P}_{\chi z}$:

$$\hat{\mathbf{z}}_{\mathbf{k}} = \sum_{i=0}^{2n} \omega_m^{[i]} \mathbf{z}_{\mathbf{k}}^{[i]}$$

$$\mathbf{P}_{zz} = \sum_{i=0}^{2n} \omega_c^{[i]} \left(\mathbf{z}_{\mathbf{k}}^{[i]} - \hat{\mathbf{z}}_{\mathbf{k}} \right) \left(\mathbf{z}_{\mathbf{k}}^{[i]} - \hat{\mathbf{z}}_{\mathbf{k}} \right)^T$$

$$\mathbf{P}_{\chi z} = \sum_{i=0}^{2n} \omega_c^{[i]} \left(\chi_{\mathbf{k}}^{[i]} - \hat{\chi}_{\mathbf{k}}^- \right) \left(\mathbf{z}_{\mathbf{k}}^{[i]} - \hat{\mathbf{z}}_{\mathbf{k}} \right)^T$$

Correction:

1. Compute the innovation $\mathbf{v}_{\mathbf{k}}$ from the current and predicted measurement $\mathbf{z}_{\mathbf{k}}$, $\hat{\mathbf{z}}_{\mathbf{k}}$, respectively:

$$\mathbf{v}_{\mathbf{k}} = \mathbf{z}_{\mathbf{k}} - \hat{\mathbf{z}}_{\mathbf{k}}$$

2. Update Kalman gain matrix:

$$\mathbf{K}_{\mathbf{k}} = \mathbf{P}_{\chi z} \mathbf{P}_{zz}^{-1}$$

3. Update the posterior mean and covariance ($\hat{\chi}_{\mathbf{k}}$, $\mathbf{P}_{\chi\chi}$):

$$\hat{\chi}_{\mathbf{k}} = \hat{\chi}_{\mathbf{k}}^- + \mathbf{K}_{\mathbf{k}} (\mathbf{z}_{\mathbf{k}} - \hat{\mathbf{z}}_{\mathbf{k}})$$

$$\mathbf{P}_{\chi\chi} = \mathbf{P}_{\chi\chi}^- - \mathbf{K}_{\mathbf{k}} \mathbf{P}_{zz} \mathbf{K}_{\mathbf{k}}^T$$

return ($\hat{\chi}_{\mathbf{k}}$, $\mathbf{P}_{\chi\chi}$)

Compute the Degree of Belief (Bel):

$$\text{Bel}_{\mathbf{k}} = 1 - \|\mathbf{P}_{\chi\chi}\|_2^{0.5}; \quad 0 \leq \text{Bel} \leq 1$$

end function

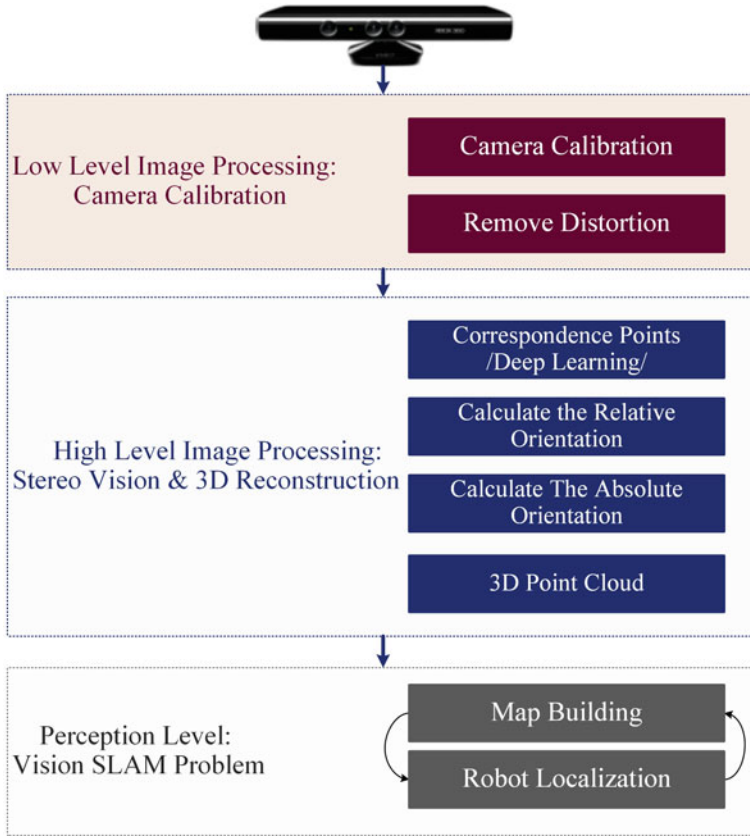


Fig. 9.2 Stereo vision 3D reconstruction stage

always intersect due to erroneous correspondences, the midpoint method or least square based method is proposed to estimate the intersection. Then to avoid the drifting problem, UKF is employed to refine both the camera pose and 3D points by minimizing re-projection errors [26].

9.3.1 Perspective Projection and Collinearity Constraint

The process by which the 3D objects are mapped onto an image by a camera is approximated by *collinearity constraint* [13].

Figure 9.3 shows the perspective projection geometry.

As seen from Fig. 9.3, light falling on the image plane is assumed to have passed through a small pinhole. Therefore, each object point P_ω maps to a single point on the image plane P_u . Three-coordinate systems are necessary to define a perspective camera model [23]: (a) The 3D world coordinate system $\{W\}$. (b) The camera

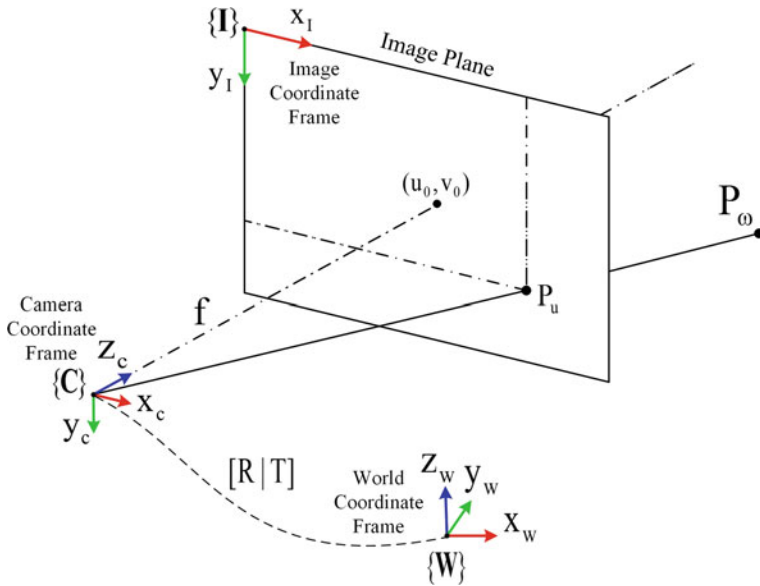


Fig. 9.3 Perspective projection model

coordinate system $\{C\}$; it is attached to the projection center of the camera. The sensor plane is parallel to its xy plane and displaced in a positive z direction. This axis pierces the image plane at the principal point (u_0, v_0) which acts as the origin of the image plane. (c) The 2D image coordinate system $\{I\}$; its origin lies at the upper left corner of the image. Two sets of parameters are used for perspective camera modeling [13]:

- **Extrinsic Parameters (Extrinsics):** These parameters describe the camera pose in the environment. Extrinsics contain six parameters of the exterior orientation (EO) of the projection center (i.e., three parameters for the translation and three other parameters for the rotation). They all vary with the camera motion in the environment.
- **Intrinsic Parameters (Intrinsic):** These parameters model the camera physics and describe the interior orientation (IO) of the camera. The intrinsic parameters are determined by calibration and are usually fixed. The parameters are now in place to define perspective projection mathematically. The mapping with an ideal perspective camera can be decomposed into two steps [23]:
 1. Exterior orientation: Given the 3D object's position in world frame ${}^wP_\omega = [{}^w x_\omega \ {}^w y_\omega \ {}^w z_\omega]^T$, the 3D object's position with respect to the camera frame using homogeneous notation ${}^C T_{W4 \times 4}$ (rotation matrix ${}^C R_{W3 \times 3}$ and translation matrix ${}^C D_{W3 \times 1}$) is given by (9.27):

$${}^C\mathbf{P}_\omega = {}^C\mathbf{T}_W {}^W\mathbf{P}_\omega \quad (9.27)$$

2. Interior orientation: A projection from camera frame to image frame using the calibration matrix κ and intrinsic parameters (the focal length \mathbf{f} , the horizontal and vertical scale vector $\mathbf{k}_u, \mathbf{k}_v$) is given by (9.28):

$$\kappa = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (9.28)$$

where $\alpha_u = -k_{uf}$, $\alpha_v = -k_{vf}$. If the shear parameter s and the scale difference m are present, they amount to an affine distortion of the image frame. It is useful to model the distortions as corrections $\Delta u, \Delta v$ of the image coordinates of a perspective camera, and the calibration matrix becomes (9.29):

$$\kappa = \begin{pmatrix} \alpha_u & s\alpha_v & u_0 + \Delta u & 0 \\ 0 & \alpha_v(1+m) & v_0 + \Delta v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (9.29)$$

If κ is known, then the camera is considered to be calibrated. The final mathematical formation of the collinearity constraint for a perspective projection with distortion from object to image frame is given by (9.30):

$${}^I\mathbf{P}_u = \kappa {}^C\mathbf{P}_\omega \quad (9.30)$$

9.3.2 Epipolar Geometry and Coplanarity Constraint

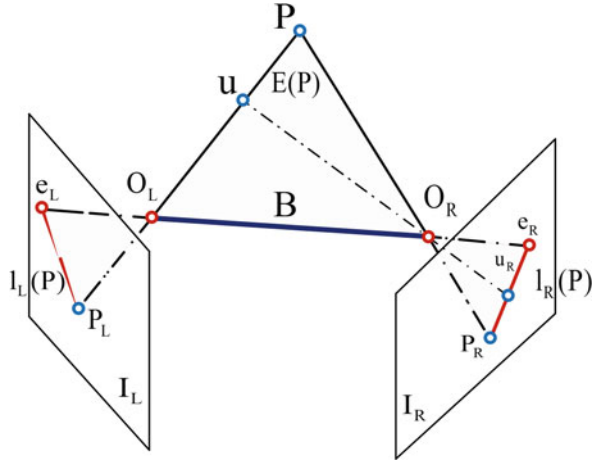
A 3D measurement cannot be derived from a single image of an unknown scene, because the depth along the Z axis is lost during projection. The principle to solve this problem is to measure the corresponding points acquired from two different viewpoints and reconstruct the 3D coordinates via triangulation. Some of these points are considered as control points [27]. Now, two problems need to be solved:

1. Determination of the image pair orientation (relative and absolute orientations).
2. Reconstruction of the 3D scene coordinates.

Figure 9.4 shows what is known as epipolar geometry. \mathbf{e}_L is the *epipole* which is the image of the right camera center in the left camera. \mathbf{e}_R is the epipole of the left camera center in the right camera. The plane formed by \mathbf{P} and the two camera centers O_L, O_R is the *epipolar plane*.

This plane intersects the image planes in the *epipolar line* l_L, l_R ; these lines can be used for matching points, and \mathbf{B} is the baseline, which is the distance between the projection center, O_R and O_L . All epipolar lines converge at the epipole. First,

Fig. 9.4 Epipolar geometry



it seems that the matching process requires cross image searching, but epipolar restrictions reduce this search to a single line [13]. The two projection rays must be coplanar because they intersect in the 3D point \mathbf{P} . For any corresponding point \mathbf{P} , triple product coplanarity can be expressed by (9.31):

$$[O_L P_L \ O_L O_R \ O_R P_R] = 0 \tag{9.31}$$

That is, the three rays are in one plane. The third dimension is extracted from a pair of images using the coplanarity constraint given for the uncalibrated camera as follow (9.32):

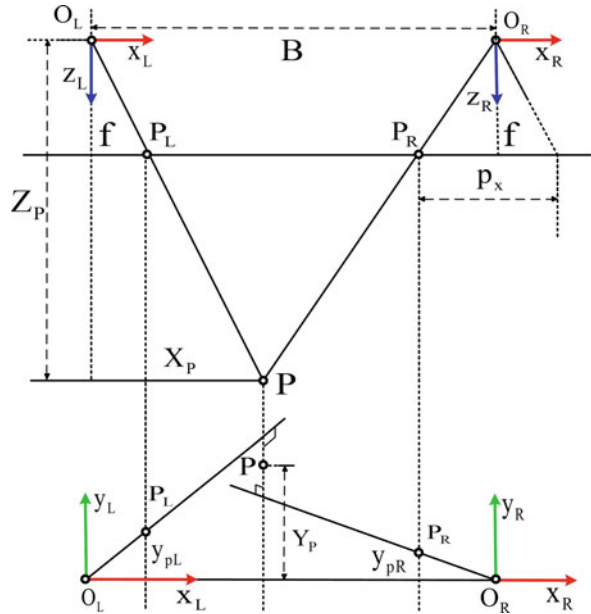
$$p_R^T \underbrace{\kappa_R^{-T} \mathbf{R}_R^{-T} \mathbf{S}_b \mathbf{R}_L^{-1} \kappa_L^{-1}}_{\mathbf{F}} p_L = 0 \tag{9.32}$$

where \mathbf{S}_b is a skew-symmetric matrix resulting from the triple product given in (9.31), κ_R, κ_L are the right and left calibration matrix, and $\mathbf{R}_R, \mathbf{R}_L$ are the rotation matrix of the right and left camera. The *fundamental matrix* $\mathbf{F}_{3 \times 3}$ sums up everything that can be known about the relationship between two uncalibrated cameras. Using $\mathbf{F}_{3 \times 3}$, it is possible to calculate the positions of the epipoles and the epipolar line in one image associated with a point in the other one. An alternative form for the epipolar geometry in case of the calibrated camera using the *essential matrix* $\mathbf{E}_{3 \times 3}$ is given by (9.33):

$$p_R^T \underbrace{\mathbf{R}_R \mathbf{S}_b \mathbf{R}_L^T}_{\mathbf{E}} p_L = 0 \tag{9.33}$$

The depth of the 3D point is calculated using the triangulation principle as shown in Fig. 9.5.

Fig. 9.5 Stereo vision triangulation [13]



The 3D coordinates of a landmark can be computed from two matched points in the left and right images by (9.34):

$$X = x_R \frac{B}{p_x}, Y = \frac{y_L + y_R}{2} \frac{B}{p_x}, Z = f \frac{B}{p_x} \tag{9.34}$$

where $x_{R, L}$, $y_{R, L}$ are the interest coordinate points in the images, X, Y, Z represent the object coordinates in the world frame, B is the baseline, and f is the focal length. The x -parallax $p_x = x_R - x_L$ is the distance between identical pixels when two images are mounted on top of each other [13].

9.4 Uncertainties in Stereo Vision System

Uncertainties are always present in the image acquisition and processing steps. Images are distorted due to various types of random noises such as Gaussian noise, Poisson noise, Quantization noise, Salt and paper noise, etc. These noises may be introduced from noise sources, for example, inaccurate image capturing devices like cameras, misaligned lenses, weak focal length, faulty memory location, etc. [28]. There are two main error categories, namely deterministic and nondeterministic [29]. The camera intrinsic parameter uncertainties are deterministic since the camera is supposed to be calibrated. In a stereo vision system with parallel optical axis as shown in Fig. 9.6, the epipolar constraint reduces to check that both the features

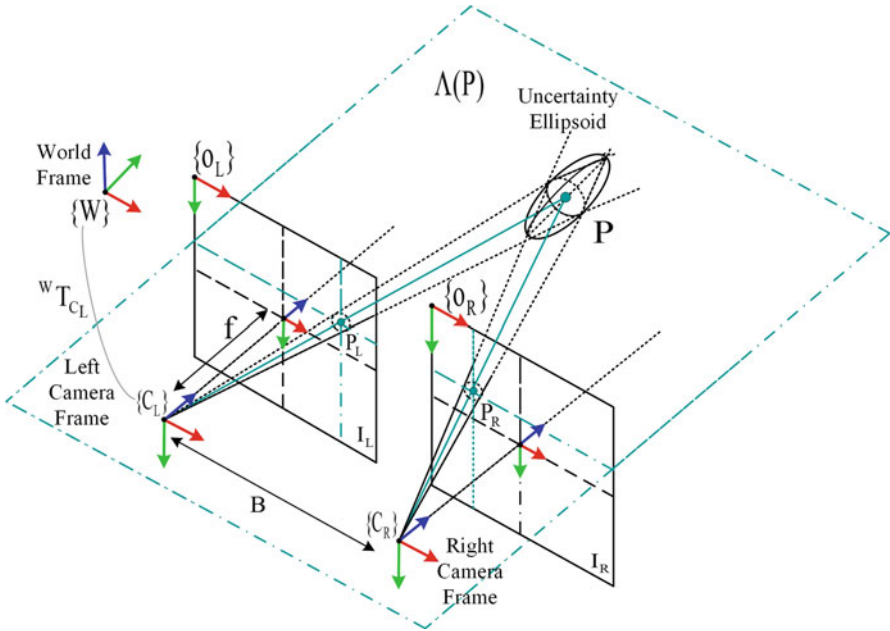


Fig. 9.6 Stereo vision uncertainty

are in the same row of the image. Consider the uncertainty in the 3D landmark position due to errors in the image quantization and in the feature detection process. Once the matching has been established, the most likely 3D coordinates of the landmark are estimated by projecting them back to the environment [30]. Refer to Eq. (9.34), error in the variables x , y , p_x are usually modeled as uncorrelated zero-mean Gaussian random variables [13]. Using the first-order error propagation to approximate the distribution of the variables in (9.34) as multivariable Gaussian, the following covariance matrix for the x , y , z coordinates has been obtained (9.35):

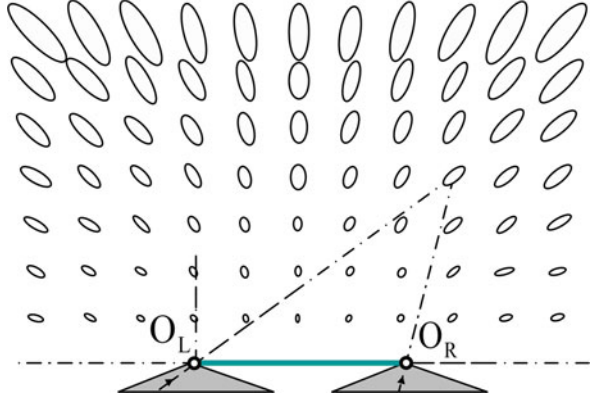
$$\Sigma \approx \mathbf{J} \text{diag}(\sigma_x^2, \sigma_x^2, \sigma_x^2) \mathbf{J}^T \tag{9.35}$$

where \mathbf{J} represents the Jacobian matrix of the functions in (9.34), and $(\sigma_x^2, \sigma_x^2, \sigma_x^2)$ are the variances of the corresponding variables.

The theoretical precision of 3D points depends on the uncertainty of the relative orientation and the uncertainty of measured corresponding points. Assume that the uncertainty of the relative orientation is negligible. By variance propagation using (9.36)

$$\sigma_X^2 = \frac{Z}{p_x} \sigma_{p_x} = \frac{fB}{p_x^2} \sigma_{p_x} = \frac{Z^2}{fB} \sigma_{p_x} = \frac{Z}{f} \frac{1}{B/Z} \sigma_{p_x} \tag{9.36}$$

Fig. 9.7 Stereo vision



where Z coordinate is the distance of the point P from the principal plane, B is the baseline, f is the focal length, p_x is the x-parallax, and σ_{p_x} is the standard deviation of the point P . The result shows the uncertainty proportional to Z^2 of the distance from the baseline B for a given geometry point P as shown in Fig. 9.7.

9.5 Examples

9.5.1 Pose Tracking Using UKF and Stereo Vision

An important improvement of the surgical robot is to extract pose information about the robot relative position to the patient. The absolute pose of the surgical robot relative to the patient cannot be observed directly due to the highly dynamic nature of the operating environment and uncertainties in the robot kinematics model [31]. To solve this problem, VSLAM is applied using the endoscopic stereo camera to estimate the robot motion. In many applications, KF framework is used to estimate the motion of the target object from the previous frame to the new frame [32]. In this example, a UKF approach is used to estimate the pose of the robot. Assuming that the feature points are observable throughout the sequence, the formulation of the UKF is as follows:

The state vector \mathbf{X}_k is given by (9.37):

$$\mathbf{X}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ z_k \ \dot{z}_k \ \alpha_k \ \dot{\alpha}_k \ \beta_k \ \dot{\beta}_k \ \varphi_k \ \dot{\varphi}_k]^T \tag{9.37}$$

where $x_k, y_k, z_k, \alpha_k, \beta_k, \varphi_k$ are the object's pose and orientation along the $x, y,$ and z axes, respectively, and $\dot{x}_k, \dot{y}_k, \dot{z}_k, \dot{\alpha}_k, \dot{\beta}_k, \dot{\varphi}_k$ are their corresponding velocities.

The dynamic system equation is given by (9.38):

$$\begin{aligned} \mathbf{X}_k &= \mathbf{A}\mathbf{X}_{k-1} + \boldsymbol{\omega}_k \\ \mathbf{A} &= \text{diag} \left[\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \right] \end{aligned} \quad (9.38)$$

where T_s is the sample time, and $\boldsymbol{\omega}_k$ is zero-mean Gaussian noise.

The nonlinear measurement model is defined as (9.39):

$$\mathbf{z}_k = h(\mathbf{X}_k) + \mathbf{v}_k \quad (9.39)$$

where \mathbf{v}_k is a $4m \times 1$ zero-mean Gaussian noise vector imposed on the images captured. m is the number of feature points extracted from the tracked robot. $h(\mathbf{X}_k)$ is the $4m \times 1$ output stereo image pair point transfer function. The estimated coordinates of the feature points at the sample time k is given by (9.40):

$$h(\mathbf{X}_k) = \left[u_{1,k}^L \ v_{1,k}^L \ \cdots \ u_{m,k}^L \ v_{m,k}^L \ \cdots \ u_{1,k}^R \ v_{1,k}^R \ \cdots \ u_{m,k}^R \ v_{m,k}^R \right]^T \quad (9.40)$$

The corresponding points have the following coplanarity constraint (9.41):

$$p_R^T \mathbf{E} p_L = 0 \quad (9.41)$$

The standard perspective projection for a single feature point is given by (9.42):

$$\mathbf{z}_k = \begin{bmatrix} u_k^L \\ v_k^L \\ u_k^R \\ v_k^R \end{bmatrix} = \begin{bmatrix} f x_x / z_x \\ f y_x / z_x \\ f (B - x_x) / z_x \\ f y_x / z_x \end{bmatrix} \quad (9.42)$$

where $\mathbf{z}_k = [u_k^L \ v_k^L \ u_k^R \ v_k^R]^T$ are the measurement pixels in the left and right images, \mathbf{B} is the baseline, and f is the focal length. The UKF algorithm can be derived using Table 9.2, giving the motion model (9.38) and the measurement model (9.39). As shown in Fig. 9.8, the robot is captured using the stereo camera with a location sensor.

The important features of the robot are then extracted and passed to the deep learning network. The robot pose with respect to the image frame is then matched with the predicted pose using UKF approach. The algorithm is initialized using epipolar geometry of the first two images and computes the essential matrix \mathbf{E} using 5-point algorithm plus robust estimator deep learning network. The initial pose parameters are then extracted from \mathbf{E} [33]. This is an initial guess of the pose and will be used in UKF approach.

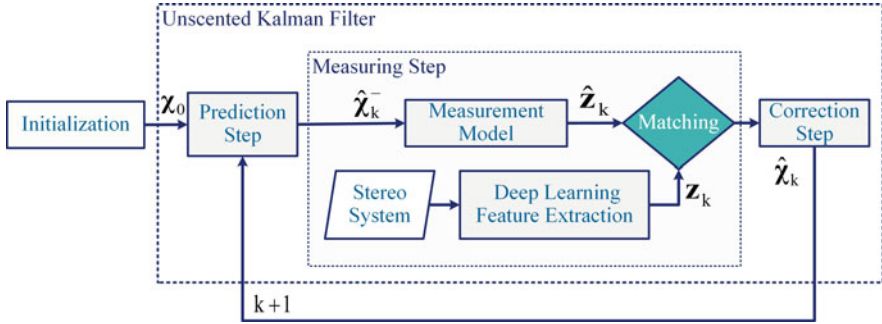
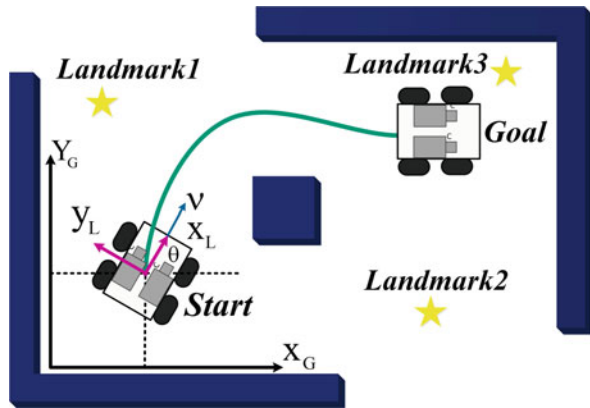


Fig. 9.8 Using UKF and stereo vision for robot tracking

Fig. 9.9 Robot localization-based landmarks [35]



9.5.2 Localization Approach-Based 2D Landmark Map

Landmarks are natural or artificial environment features which a robot can recognize from its sensory input and keep the uncertainty bounded [34]. Consider 4-WDDMR moves in the predefined environment in the global frame $\{G\}$ as shown in Fig. 9.9. Stereo vision system is used as an exterior receptive sensor to enable the robot to recognize the landmarks [36]. The predefined landmarks in the global frame $\{G\}$ help the robot to localize itself correctly. The 4-WDDMR discrete kinematic equation is given by (9.43):

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x_{k-1} + v_k T_s \cos(\theta_{k-1}) \\ y_{k-1} + v_k T_s \sin(\theta_{k-1}) \\ \theta_{k-1} + \omega_k T_s \end{bmatrix} \tag{9.43}$$

where T_s is the sample time and $f(\mathbf{x}_{k-1}, \mathbf{u}_k)$ is a nonlinear function and relates the 4-WDDMR's pose $\mathbf{x}_k = [x_k \ y_k \ \theta_k]^T$ in $\{G\}$ frame, with the input vector $\mathbf{u}_k =$

$[v_k \ \omega_k]^T$, which represents the translation and angular velocities in the robot frame $\{L\}$. However, the noisy 4-WDDMR motion model with non-additive Gaussian noise w_k is written as follows (9.44):

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \quad (9.44)$$

In the authors' previous work [37], it is assumed that landmarks have a fixed and known position (m_x, m_y) .

The measurement model of the location z_k of the landmark m_k from the viewpoint of the robot given the location of the robot x_k is defined as follows (9.45):

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{m}_k) = \begin{bmatrix} h_x(\mathbf{x}_k, \mathbf{m}_k) \\ h_y(\mathbf{x}_k, \mathbf{m}_k) \end{bmatrix} \\ &= \begin{bmatrix} (m_{xk} - x_k) \cos(\theta_k) + (m_{yk} - y_k) \sin(\theta_k) \\ -(m_{xk} - x_k) \sin(\theta_k) + (m_{yk} - y_k) \cos(\theta_k) \end{bmatrix} \end{aligned} \quad (9.45)$$

where $h(\mathbf{x}_k, \mathbf{m}_k)$ is the nonlinear measurement function for the correction stage.

9.6 Conclusion

This book chapter has provided modern and advanced strategies for image filtering and image feature extraction. Kalman filter including EKF and UKF has been described in detail and implemented for pose tracking and localization-assisted 2D landmark map of a mobile robot. This chapter has contributed to the community of mobile robot localization and mapping, with detailed information on different modern approaches for vision-based localization and mapping. It has been pointed out that UKF algorithm is superior to EKF one in terms of bias cancellation and providing higher accuracy of mobile robot pose estimation.

References

1. Goliaei, S., Ghorshi, S., Manzuri, M. T., & Mortazavi, M. (2011). A Kalman filter techniques applied for medical image reconstruction. In *8th International Multi-Conference on Systems, Signals & Devices*.
2. Ling, L. (2013). *Dense real time 3D reconstruction from multiple images*. PhD thesis, College of Science, Engineering and Health, RMIT University.
3. Saputra, M. R. U., Markham, A., & Trigoni, N. (2018). Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys*, 51, 2.
4. Klančar, G., Teslic, L., & Skrjanc, I. (2014). Mobile robot pose estimation and environment mapping using an extended Kalman filter. *International Journal of Systems Science*, 45(12), 2603–2618.

5. Basaca-Preciado, L. C., Sergiyenko, O. Y., Rodríguez-Quinonez, J. C., Garcia, X., Tyrsa, V. V., Rivas-Lopez, M., & Tabakova, I. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Lasers in Engineering*, 54, 159–169.
6. Park, J., & Lee, S. (2009). Correction robot pose for SLAM based on extended Kalman filter in rough surface environment. *International Journal of Advanced Robotic System*, 6(2), 67–72.
7. Moreno, F. A., Blanco, J. L., & Gonzalez, J. (2009). Stereo vision-specific models for particle filter-based SLAM. *Robotics and Autonomous Systems*, 57(9), 955–970.
8. Tully, S. T. (2012). *BodySLAM: Localization and mapping for surgical guidance*. PhD thesis, Garnegir Mellon University, Pittsburgh.
9. Szeliski, R. (2010). *Computer vision: Algorithms and applications*. Springer.
10. Thrun, S., Fox, D., & Burgard, W. (2006). *Probabilistic robotics*. Massachusetts Institute of Technology.
11. Merwe, R. (2004). *Sigma point Kalman Filter for probabilistic inference in dynamic state space models*. PhD thesis, Oregon Health & Science University.
12. Sibley, G., Sukhatme, G., & Matthies, L. (2006). *The iterated sigma point Kalman filter with applications to long range stereo*. Robotics Science and Systems.
13. Förstner, W., & Wrobel, B. P. (2016). *Photogrammetric computer vision statistics, geometry, orientation and reconstruction*. Cham, Switzerland: Springer.
14. Särkkä, S. (2011). *Bayesian filtering and smoothing* (Vol. 3). Cambridge University Press.
15. Corke, P. (2011). Robotics vision and control fundamental algorithms in MATLAB. In *Springer tracts in advanced robotics* (Vol. 73). Springer.
16. Haykin, S. M. (2001). *Kalman filtering and neural networks*. Wiley.
17. Joukhadar, A., & Kass Hanna, D., (2018). UKF and adaptive optimal control-based localization enhancement of 4WDDMR, ROS framework-based design and implementation. In *Cogent engineering, System and Control Research Article*.
18. Long, A. W., Wolfe, K. C., Mashner, M. J., & Chirikjian, G. S. (2013). *The banana distribution is Gaussian: A localization study with exponential coordinates* (pp. 265–272). Cambridge, MA: Robotics: Science and Systems VIII; MIT Press.
19. Mahmoudi, Z., Poulsen, N. K., Madsen, H., & Jorgensen, J. B. (2017). Adaptive unscented Kalman filter using maximum likelihood estimation. *IFAC-Papers Online*, 50(1), 3859–3864. <https://doi.org/10.1016/j.ifacol.2017.08.356>.
20. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes. The art of scientific computing*. Cambridge, UK: Cambridge University Press.
21. Rodríguez-Quirfónez, J. C., Sergiyenko, O., Flores-Fuentes, W., Rivas-lopez, M., Hernandez-Balbuena, D., Rascón, R., & Mercorelli, P. (2017). Improve a 3D distance measurement accuracy in stereo vision system using optimization methods' approach. *Opto-Electronics Review*, 25(1), 24–32.
22. Hu, Y., Chen, Q., Feng, S., Tao, T., Asundi, A., & Zuo, C. (2019). A new microscopic telecentric stereo vision system—Calibration, rectification, and three-dimensional reconstruction. *Optics and Lasers in Engineering*, 113, 14–22.
23. Bergamini, M. L., Ansaldo, F. A., Bright, G., & Zelasco, J. F. (2017). Digital camera calibration, relative orientation and essential matrix parameters. *WSEAS Transaction on Signal Processing*, 13.
24. Hayet, J. B., Lerasle, F., & Devy, M. (2002). A visual landmark framework for indoor mobile robot navigation. In *International Conference on Robotics & Automation*. Washington, DC.
25. Florczyk, S. (2005). *Video-based indoor exploration with autonomous and mobile robots*. Wiley.
26. Hartmann, G., Huang, F., & Klette, R. (2013). *Landmark initialization for unscented Kalman filter sensor fusion in monocular camera localization*. Auckland, New Zealand: The University of Auckland.
27. Parnian, N., & Golnaraghi, F. (2010). Integration of multi-camera vision system and strap down inertial navigation system (SDINS) with a modified Kalman filter. *Sensors Journal*, 10(6), 5378–5394.

28. Boyat, A. K., & Joshi, B. K. (2015). A review paper: Noise models in digital image processing. *Signal & Image Processing: An International Journal (SIPIJ)*, 6(2).
29. Siegwart, R., & Noubakhsh, I. R. (2004). *Introduction to autonomous mobile robots*. Cambridge, MA, London: The MIT Press.
30. Beinhofer, M. Müller, J., Krause, A., & Burgard, W. (2013). Robust landmark selection for mobile robot navigation. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference*.
31. Haghighipanah, M., Miyasaka, M., Li, Y., & Hannaford, B. (2016). Unscented Kalman filter and vision to improve cable driven surgical robot joint angle estimation. In *IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden*.
32. Yu, Y. K., Wong, K. H., Or, S. H., & Chang, M. M. Y. (2006). Recursive camera motion estimation with trifocal tensor. *IEEE Transaction on System, Man and Cybernetics B*, 36(5), 1081–1090.
33. Yu, Y. K., Wong, K. H., Or, S. H., & Chang, M. M. Y. (2008). Robust 3D motion tracking from stereo images: A model-less method. *IEEE Transaction on Instrumentation and Measurement*, 57(3).
34. Negenborn, R. (2003). *Robot localization and Kalman Filters on finding your position in a noisy word*. MSc thesis, Utrecht University.
35. Joukhadar, A., Kass Hanna, D., Müller, A., & Stöger, C., (2017). UKF-Assisted SLAM for 4WDDMR Localization and Mapping. In *1st International Congress for the Advancement of Mechanism, Machine, Robotics and Mechatronics Sciences, Beirut, Lebanon*, 17–19 October.
36. Kelly, J., & Sukhatme, G. S. (2009). *Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration*. Korea: CIRA.
37. Kass Hanna, D., & Joukhadar, A. (2015). A novel control-navigation system-based adaptive optimal controller & EKF localization of DDMR. *International Journal of Advance Research in Artificial Intelligence*, 4(25), 29–37.