Chapter 1

REVIEW ON SENSORS AND RELATED WORK

1.1 SENSORS FOR OBSTACLE DETECTION

1.1.1 Overview

Many work in obstacle detection uses range sensors such as radar, laser range finder, LIDAR, sonar. [reference goes here]. Radar and laser range finder provides only point measurement at a given position and orientation. To acquire a full 3D range map of a scene, mechanical scanning mechanisms are required, which limits the data acquisition rate of these device. LIDAR operate in the same manner as laser range finder, except with the scanning mechanism built in. These sensors usually have high power requirement and mass, and may not be suitable for small and mid size UAV. Sonar is usually used in indoor or under water applications, and have wide beam profile which make it difficult to identify the origin of return signal, and results in low resolution range map. 3D flash LIDAR is capable of acquire 3D range measurement simultaneously by illuminating the entire field of view of the camera with a single laser, and capturing the reflected laser with a 2D imaging sensor [reference; wikipedia].

However, its high cost has limited its use in commercial application.

In recent years, many researches use optical sensor as a passive range sensor for its low weight, low cost. With the help of computer vision technology, optical sensors have been successfully used for range mapping and obstacle detection in a number of platforms [references]. There are several types of configuration in using optical sensor for range mapping: monocular, binocular, or multi-camera. Since optical sensors are bearing only sensors, the principle of range measurement is through triangulation a common scene point in two or more images captured. For binocular camera setups, two cameras are placed apart from each other with their relative geometry known and captures images simultaneously. If the position of a scene point can be accurately found in the images by both cameras, its distance can be calculated by using the difference in position of the projected point in images, and the separation of the cameras.

- Radar, sonar, laser range finder, 3D flash lidarvs. optical sensors
  - Radar, laser range finder have high power requirement and mass
  - Depth measurement can be obtained through optical sensors, which are inexpensive and light weight
  - Depth maps of a 3-D scene can be computed from a single pair of stereo camera. Stereo processing can require significant computational effort
- Monocular camera characteristic:
  - bearing-only sensor, which provide the measurement on the direction of the feature, and not the range. Other sensors, such as radar, are range and bearing sensors.

1.1.2 Monocular Vision and Binocular Vision

- Optical flow vs. feature detection and tracking
- The correspondence problem
- Initialization problem (addressed by Inverse depth parameterization)
- Lack of scale information of overall map -> must work with other sensors which provide robot motion measurement.

1.1.3 Limitation of Optical Sensor in Recursive Algorithm

- Error Accumulation over Iterations
  - Feature quality Decreases over Iterations

1.1.4 GPS and IMU

GPS and IMU are generally available on UAVs. These sensors provide a measurement on the robot motion. Odometry can provide the scale information which is missing in the bearing only measurement. Furthermore, odometry provides some prior information on the robot motion which can help to disambiguate the solution.

1.2 SLAM as A Sensor Fusion Framework

An essential aspect of autonomy for a mobile robot is the capability to determine its location. This capability is known as localization. Localization is typically a prerequisite for accomplishing real tasks, whether it is exploration, navigation toward a known goal, transportation of material, construction or site preparation. In many applications, the mobile robot has an a priori map. Given a map, the robot may localize by matching current sensor observations to features in the map. Given enough features and an unambiguous geometry, the pose of the robot can be determined or at least narrowed down to a set of possible locations.

Usable maps do not always exist, and it is not always possible to have accurate externally referenced pose estimates. If an a priori map is not available, the robot may need to construct one. With a precise, externally referenced position estimate from GPS or similar means, the robot can take its sensor observations, reference the observations to its current pose, and insert features in the map in the appropriate places. Without maps or externally referenced pose information, the robot must produce its own map and concurrently localize within that map. This problem has been referred to as concurrent localization and mapping (CLM) and simultaneous localization and mapping (SLAM).

1.2.1 Recursive Probabilistic Estimation using Extended Kalman Filter

Kalman Filter

The Kalman filter [**?**]published by R. E. Kalman in 1960 is a very powerful recursive data processing algorithm for dynamic stochastic processes. The filter find extensive use in control and navigation application for its ability of estimating past, present and even future state. It is an attractive candidate for data fusion framework as it can process all available measurements including previous knowledge of the process, regardless of their precision, to estimate the current value of the variable of interest. Given a dynamic process that satisfy the assumptions that Kalman filter is based on, the filter is the optimal algorithm in minimzing the mean of squared error of the state variable. This section briefly summerized assumption and formation of Kalman filter that's described in detail in [**?**][Sorenson70; Gelb74; Grewal93; Lewis86; Brown92]. A more intuitive introduction can be found in chapter 1 of [Maybeck79]

The Kalman filter has three assumptions. 1) The system model is linear. The linearity is deisred in that the system model is more esily manipulated with engineering tool. When nonlinearities do exist, the typical approach is to linearize system model at some nominal points. 2)The noise embedded in system control and measurement is white. This assumption implies that the noise value is not correlated in time, and has equal power in all frequency. 3)The probability density function (PDF) of system and measurement noise is Gaussian. A Gaussian distribution is fully represented by the first and second order statistic (mean and variance) of a process. Most other densities require endless number of orders of statistic to describe the shape fully. Hence, when the probability density function of a noise process is non-Gaussian, the Kalman filter that propagates the first and second order statistic only include some of the information of the PDF, instead of all, as would be the case with Gaussian noise.

Kalman Filter Models The Kalman filter requires two models. The process model define a discrete-time controlled process by a linear stochastic difference equation. The $n \times n$ matrix A relates the state variables x_{k-1} in previous time step $k-1$ to the state variabl x_k in the current time step k. The matrix B relates the optional control input μ to the state variables x. Given measurement vecor z_k of size $1 \times m$, the measurement model relates the state variables to the measurements by matrix H of size $n \times m$. The random variable w and v represent the uncertainty or noise of the process model, and measurement. w and v are assumed to be unrelated to each other, and has Gaussian distribution with covariance Q and R.

$$\text{Process Model: } x_k = Ax_{k-1} + B\mu_{k-1} + w_{k-1} \tag{1}$$

$$\text{Measurement Model: } z_k = Hx_k + v_k \tag{2}$$

The Algorithm The Kalman filter operates in prediction and correction cycle after being initialized 1, with the state vector estimate $\hat{x}_k^-, \hat{x}_k^+$ contains the variable of interest at time step k, and state covariance matrix P_k^-, P_k^+ representing the error covariance of the estimate. The superscript - indicate the estimate is a priori (or predicted) estimate, and + indicate the estimate is a posteriori (or corrected) estimate. The equations used for prediction and correction are listed in 1. In prediction, an estimate of the state variables are made based on the known knowledge of the process (the process model). Since there are always unknown factor not fully described by the process model, the error of the estimate almost always increase in the prediction. During correction, a series of calculation were carried out to correct the a priori estimate. First, the predicted measurement $H\hat{x}_k^-$ are compared to the new measurement z_k. Their difference $z_k - H\hat{x}_k^-$ is called the measurement innovation, or residual. Next, the amount of residual is weighted by the Kalman gain K, and added to $\hat{x}_k^-$ as correction. The Kalman gain is fomulated so that it minimize the a posteriori error covariance matrix P_k^+.

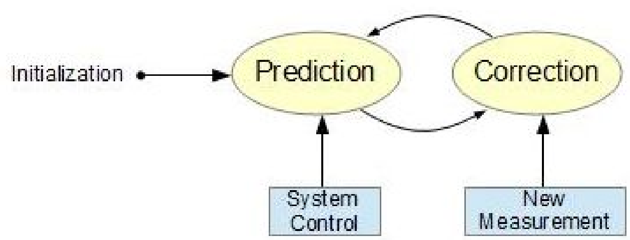

Figure 1: Kalman Operation Flow Diagram

Table 1: Kalman Filter Operation Equations

Prediction	$\hat{x}_k^- = A\hat{x}_{k-1}^+ + B\mu_{k-1}$	3
	$P_k^- = AP_{k-1}^+A^T + Q$	4
Correction	$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1}$	5
	$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$	6
	$P_k^+ = (I - K_kH)P_k^-$	7

Extended Kalman Filter For a discrete-time controlled process, or its relationship with the measurements are non-linear, a Kalman filter must be linearized about the estimated trajectory, and is referred to as an extended Kalman filter or EKF. A process with state vector x and measurement z that is governed by non-linear stochastic difference equation has process and measurement model

$$x_k = f(x_{k-1}, u_{k-1} + w_{k-1}), \tag{8}$$

$$z_k = h(x_k + v_k), \tag{9}$$

where the random varialbe w_k and v_k represent the process and measurement noise with variance Q and R. The the Kalman filter operation equations are given in table 2,

Table 2: Extended Kalman Filter Operation Equations

Prediction	$\hat{x}_k^- = f(\hat{x}_{k-1}^+, \mu_{k-1}, 0)$	10
	$P_k^- = A_kP_{k-1}^+A_k^T + W_kQ_{k-1}W_k^T$	11
Correction	$K_k = P_k^- H_k^T(H_kP_k^- H_k^T + V_kR_kV_k^T)^{-1}$	12
	$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$	13
	$P_k^+ = (I - K_kH)P_k^-$	14

where (subscript k omitted)

- A is the Jacobian matrix of partial derivatives of f with respect to x,

$$A_{[i,j]} = \frac{\partial f_i(\hat{x}_{k-1}^{+}, \mu_{k-1}, 0)}{\partial x_j}$$

- W is the Jacobian matrix of partial derivatives of f with respect to w,

$$W_{[i,j]} = \frac{\partial f_i(\hat{x}_{k-1}^{+}, \mu_{k-1}, 0)}{\partial w_j}$$

- H is the Jacobian matrix of partial derivatives of h with respect to x,

$$H_{[i,j]} = \frac{\partial h_i(\hat{x}_{k}^{-}, 0)}{\partial x_j}$$

- V is the Jacobian matrix of partial derivatives of h with respect to v,

$$V_{[i,j]} = \frac{\partial h_i(\hat{x}_{k}^{-}, 0)}{\partial v_j}$$

Note that when w and v directly describe the noise of state vector and measurement, the table 2 is the same as table 1.

Tuning

Properties of SLAM

Needs editing.Directly from papers.

Dissanayake proved three important convergency properties of the EKF solution to SLAM, namely that: (1) the determinant of any submatrix of the map covariance matrix decreases monotonically as observations are successively made; (2) in the limit as the number of observations increases, the landmark estimates become fully correlated and (3) in the limit the covariance associated with any single landmark location estimate reaches a lower bound determined only by the initial covariance in the vehicle location estimate at the time of the first sighting of the first landmark.

The properties imply:

- The entire structure of the SLAM problem critically depends on maintaining complete knowledge of the cross correlation between landmark estimates. Minimizing or ignoring cross correlations is precisely contrary to the structure of the problem. (Early EKF for OD work eliminate the cross correlations between features and veichel pose in an attempt to reduce computation complexity.)

- As the vehicle progresses through the environment the errors in the estimates of any pair of landmarks become more and more correlated, and indeed never become less correlated.

- In the limit, the errors in the estimates of any pair of landmarks become fully correlated. This means that given the exact location of any one landmark, the location of any other landmark in the map can also be determined withabsolute certainty.

- As the map converges in the above manner, the error in the absolute location of every landmark (and thus the whole map) reaches a lower bound determined only by the error that existed when the first observation was made (Initialize the parameters using 1st frame as coordinate origin with minimum variance - Algorithm initialization).

It is important to note that these theoretical results only refer to the evolution of the covariance matrices computed by EKF in the ideal linear case. They overlook the fact that given that SLAM is a nonlinear problem, there is no guarantee that the computed covariance will match the actual estimation erros which is the true SLAM consistency issue.

Linearization Error and Consistency

Many research report filter divergence due to linearization error. Literature review here: As defined in [**?**], a state estimator is consistent if the estimation errors (i) are zero-mean, and (ii) have covariance matrix smaller or equal to the one calculated by the filter.

Huang investigate further on properties and consistency of nonlinear two-dimentional EKF based SLAM problem, and conclude:

- Most of the convergence properties in [3] are still true for the nonlinear case provided that the Jacobians used in the EKF equations are evaluated at the true states.
- The main reasons for inconsistency in EKF SLAM are due to (i) the violation of some fundamental constraints governing the relationship between various Jacobians when they are evaluated at the current state estimate, and (ii) the use of relative location information from robot to landmarks to update the absolute robot and landmark location estimates.

The robot orientation uncertainty plays an important role in both the EKF SLAM convergence and the possible inconsistency. In the limit, the inconsistency of EKF SLAM may cause the variance of the robot orientation estimate to be incorrectly reduced to zero.

Linearization error can be interpreted as error resulted from calculating the Jacobian at the estimated state (wrong state) instead of the true state.

Camera Centric Coordinate System

SLAM for Large Scale Maps

List of References

[1] "Athena 111m integrated flight control system." http://www.rockwellcollins.com/sitecore/content/Data/Products/Controls/Flight_Controls/At

[2] "CGIAR-CSI SRTM 90m DEM digital elevation database." http://srtm.csi.cgiar.org/.

[3] "World geodetic system - wikipedia, the free encyclopedia." https://en.wikipedia.org/wiki/World_Geodetic_System.

[4] "pyproj - python interface to PROJ.4 library - google project hosting." http://code.google.com/p/pyproj/.