

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

## 4,800

Open access books available

## 122,000

International authors and editors

## 135M

Downloads

Our authors are among the

## 154

Countries delivered to

## TOP 1%

most cited scientists

## 12.2%

Contributors from top 500 universities

**WEB OF SCIENCE™**Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

## Interested in publishing with us? Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)

## General Concept of 3D SLAM

Peter Zhang, Evangelous Millos & Jason Gu  
Dalhousie University  
Canada

Simultaneous localization and mapping (SLAM) is a process that fuses sensor observations of features or landmarks with dead-reckoning information over time to estimate the location of the robot in an unknown area and to build a map that includes feature locations. In this chapter, a general model and its related solving algorithm for 3D SLAM are established. The method can be used for all of the situations in the mobile robot community. An underwater mobile robot is used as an example.

This chapter is organized as follows: Section 1 is the problem definition; Section 2 establishes all the models for 3D SLAM, including the robot process model, the landmark model, and the measurement model; Section 3 is the method for data association; Section 4 presents the algorithms to solve the SLAM; section 5 describes the multi-sensor related issues based on the underwater mobile robot cases; and Section 6 is the globally-consistent 3D SLAM for mobile robot in real environment.

### 1. Problem Definition

Assuming a 3D environment with randomly distributed landmarks and an autonomous mobile robot equipped with sensors (stereo camera, laser range finder, or sonar) which will move in this environment, by providing some proper input (robot speed and orientation), we need to determine the robot pose (position and orientation) and the position of detected landmarks during the robot navigation. Because of measurement noise and robot input noise, it is very difficult to compute a deterministic value for the robot pose and landmark position. We can only estimate their approximate value by using algorithms such as the Kalman filter, the Particle filter, and the Unscented Kalman filter. By using these algorithms, it is also possible to calculate the confidence of the estimation. In some areas of the robot's working environment, significant landmarks are sparse, especially in the underwater environment. A robot equipped with only one type of sensor may not obtain sufficient effective measurements, which would greatly affect the accuracy of the robot pose; therefore, more than one sensor will be used for the robot navigation in a real application. In this thesis, the SLAM problem in the 3D environment will be solved with multiple heterogeneous sensors. A general strategy will be proposed and related algorithms will be developed.

## 2. Models for 3D SLAM

### 2.1. Robot Process Model

Robot process model is a dynamic differential equation to describe the movement of a robot in a given environment and system input. It is related to the robot pose. The robot pose can be determined by its position and orientation. In a global coordinate system OXYZ, a robot position ( $p_v$ ) is expressed by  $(x, y, z)^T$ , and its orientation can be expressed by Euler angles, rotation matrix, axis and angle, or quaternion. From any one of the orientation representations, it is possible to compute the other representations. For simplicity, Euler angles are selected as a robot orientation state vector. Therefore, the state vector of the robot  $X_v$  can be expressed as

$$X_v = \begin{bmatrix} p_v^T \\ \theta_v^T \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \quad (1)$$

where T is the transpose of a matrix and assuming that the robot moves relative to its current pose with speed  $v$  and changes direction with Euler angles  $(\delta\theta_x, \delta\theta_y, \delta\theta_z)$ , the input to the robot can be expressed by

$$U = \begin{bmatrix} v \\ \delta\theta_x \\ \delta\theta_y \\ \delta\theta_z \end{bmatrix} \quad (2)$$

where  $v$  is the robot speed in scalar, and the direction of the speed is always in the robot's forward pointing axis of its body. In order to simplify its implementation, the Euler angles need to be expressed in the form of a rotation matrix  $M_v$

$$M_v = R_z(\theta_z) \cdot R_y(\theta_y) \cdot R_x(\theta_x) \quad (3)$$

where  $R_z$ ,  $R_y$ , and  $R_x$  are the rotation matrices which are the rotation around the  $z$ ,  $y$ ,  $x$ -axis, respectively, in right hand coordinate system with positive angle  $\theta_z, \theta_y, \theta_x$ , the positive angle is at counter-clockwise direction. Then, the robot process model can be expressed as

$$\theta_u(k+1) = \begin{bmatrix} \theta_x(k+1) \\ \theta_y(k+1) \\ \theta_z(k+1) \end{bmatrix} = \begin{bmatrix} f_1(\theta_x(k), \delta\theta_x, \delta\theta_y, \delta\theta_z) \\ f_2(\theta_y(k), \delta\theta_x, \delta\theta_y, \delta\theta_z) \\ f_3(\theta_z(k), \delta\theta_x, \delta\theta_y, \delta\theta_z) \end{bmatrix} \quad (4)$$

and

$$P_u(k+1) = \begin{bmatrix} x(k+1) \\ y(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix} + M_u(k) \begin{bmatrix} \cos(\alpha) \\ \cos(\beta) \\ \cos(\gamma) \end{bmatrix} v \delta t \quad (5)$$

where  $\delta t$  is the sampling time,  $M_v(k)$  is the rotation matrix, which corresponds to the Euler angles  $\theta_x(k), \theta_y(k), \theta_z(k)$  at time  $k$ . In Equation(4), the angle  $\theta_v(k+1)$  corresponds to the matrix  $M_v(k+1)$ , which has following equation

$$M_v(k+1) = M_v(\delta\theta) \cdot M_v(k) \quad (6)$$

where  $M_v(\delta\theta)$  is a matrix which corresponds to the Euler angle  $\delta\theta$  and in Equation.(5), And the  $\alpha, \beta, \gamma$  are direction angles corresponding to the Euler angles,  $\theta_x(k), \theta_y(k), \theta_z(k)$ .

By combining the Equation(4) and (5), the process model can be written as a non-linear equation

$$X_u(k+1) = F(X_u(k), U(k) + \mu(k) + \omega(k)) \quad (7)$$

where  $\mu(k)$  the input is noise, and  $\omega(k)$  is the process noise, at the sample time  $k$ . The noise is assumed to be independent for different  $k$ , white, and with zero mean and covariance  $Q_v(k)$ .

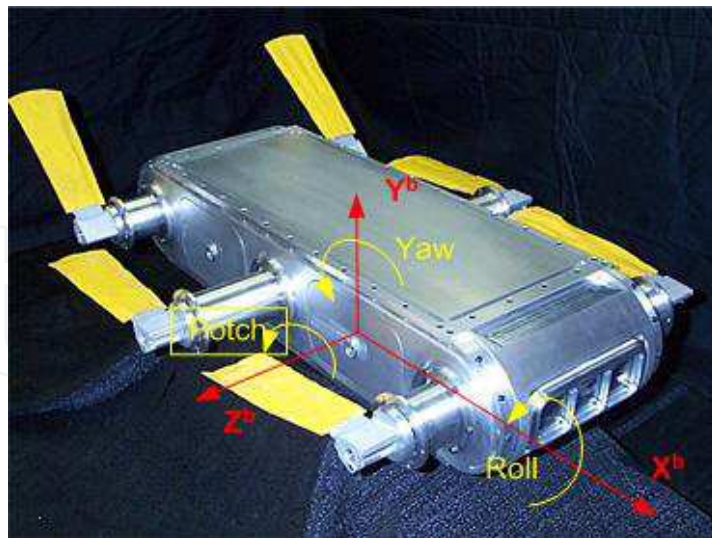


Fig. 1. Coordinate systems of an autonomous mobile robot.

## 2.2. Landmark Models

Landmarks can be classified into two types, artificial and natural. In a newly-visited natural environment, there is no artificial landmark for mobile robot navigation; therefore the natural landmarks are the only choice.

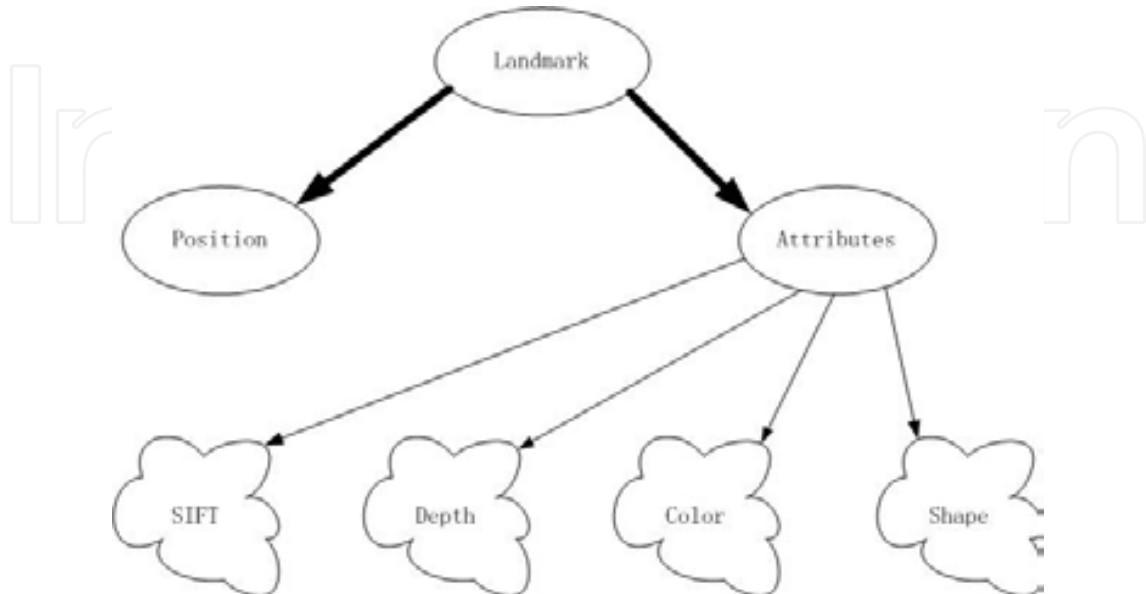


Fig. 2. General landmark expression.

A robot map consists of a set of landmarks. In order to provide enough information for robot navigation, every landmark should include position information and attribute information (Figure. 2). If the landmark position is known, a sensor's measurement of it can be used in the robot pose estimation with algorithms such as the extended Kalman filter or the particle filter. If the landmark position is unknown, an algorithm for SLAM will be applied to estimate the robot pose and landmark position by the aid of measurements. The attribute information will provide knowledge about the landmark which distinguishes it from other features, which is very useful for data association; therefore, landmark  $L_i$  can be expressed as

$$L_i = \begin{bmatrix} L_{i,position} & L_{i,attribute} \end{bmatrix} \quad (8)$$

where

$$L_{i,position} = L_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}. \quad (9)$$

During robot navigation, even though its environment is unknown, the landmarks for a map establishment are always assumed to have a static position. It is also assumed that attributes (or features) of a landmark will not change. In reality, features of a landmark may change with lighting conditions and sensor view point, therefore, the landmark  $i$  has the following evolution equation

$$L_i(k+1) = L_i(k) \quad (10)$$

where  $i = 1, \dots, m$ , which means there are  $m$  landmarks which will be used;  $k$  is the time which is used during the robot navigation.

### 2.3. Measurement Model

A mobile robot is always equipped with some type of sensors for its navigation. The sensors can obtain measurements of the relative location of the observed landmarks with respect to the robot. This observation can be expressed by a set of non-linear functions of the landmark's position relative to the robot position, which is called measurement model. Assuming the position of landmark  $i$  in the global coordinate system  $OXYZ$  is  $(x_i, y_i, z_i)$ . At time  $k$ , the robot has the pose  $X_u(k)$ . The measurement of the landmark  $i$  at this time can be computed by

$$Z_i = \begin{bmatrix} Z_{x_i}(k) \\ Z_{y_i}(k) \\ Z_{z_i}(k) \end{bmatrix} = M_v(k) \begin{bmatrix} x_i - x(k) \\ y_i - y(k) \\ z_i - z(k) \end{bmatrix}. \quad (11)$$

The observation model in the non-linear equation is

$$Z_i = h(X_v(k), x_i, y_i, z_i) + \eta(k) = h(X_v(k), L_i(k)) + \eta(k) \quad (12)$$

where  $\eta(k)$  is the observation noise, which is assumed with zero mean and covariance  $R_i$  and  $h(\cdot)$  is the non-linear measurement function.

Without loss of generality, it is assumed that the measurement from every sensor is independent. If there are  $m$  features observed at time  $k$ , the measurement model is obtained by simply stacking Equation. (12) as

$$Z(k) = \begin{bmatrix} Z_1(k) \\ Z_2(k) \\ \dots \\ Z_m(k) \end{bmatrix} = H(X_v(k), L(k)) + \eta(k). \quad (13)$$

An instance of a robot and five landmarks in 3D space is shown in Figure 3. When the robot moves in space, its sensor detects landmarks which are located in the view field of the sensor, then the robot pose and landmark position estimation can be performed. The estimated landmark position will be used to build a map which can be used by the robot for future navigation.



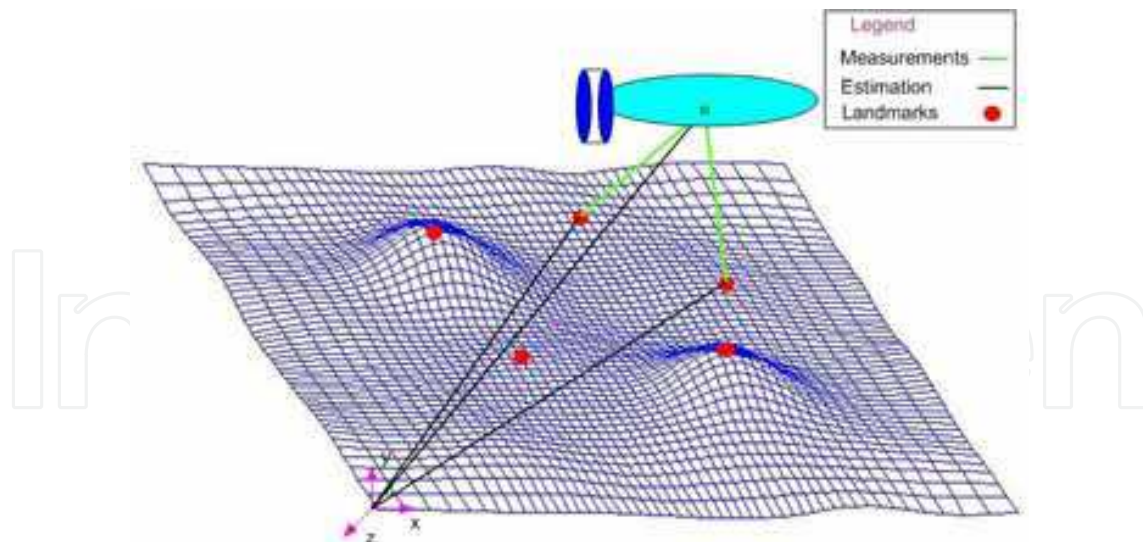


Fig. 3: An instance of a robot and features in 3D experiment case.

### 3. Data Association

There are two types of data associations - measurement between sensors received from multi-sensors, and measurement between adjacent times from a single sensor. In this thesis, only the data association between adjacent times from a single sensor will be addressed.

During robot navigation, if the sensor on the robot only observed one landmark, there would be no need for data association. Most sensors, such as camera, radar, laser, and sonar, will detect not only many real landmarks, but also many spurious landmarks; therefore, data association is a necessary step for landmark-based robot localization and object tracking.

A landmark defined in the previous Section 2.2. includes position and attributes (feature) components. If the attribute tuple is available, then data association can be implemented with this information; otherwise, maximum likelihood of measurement will be used.

#### 3.1. Data Association by Feature Attribute

Data association by using the landmark's attribute is simple. A very good example is the image feature registration with SIFT (Scale Invariant Feature Transforms) features (Se et al., 2003). SIFT features in an associated image are among the best representations for the natural unstructured environment. The SIFT features are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. The structure of the SIFT feature is as follows:

$$[u, v, \text{gradient}, \text{orientation}, \text{descriptor}_1, \dots, \text{descriptor}_M]$$

where  $M$  is the number of the descriptor in SIFT feature. The position of landmark and the position of its related SIFT features (in an image) of a landmark have a non-linear relation. If a camera's physical position and orientation are given, the position of the landmark associated with the SIFT feature can be calculated by using its associated camera model (Sonka et al., 1999).



Fig. 4. SIFT features in an image.

An example of SIFT features in an image is shown in Figure 4. The data association for SIFT features can be carried out by using their feature descriptor directly. SIFT features correspondence between two adjacent images obtained from a moving camera at different view points after the implementation of data association is displayed in Figure 5.

IntechOpen





Fig. 5. Result of SIFT feature correspondence between two adjacent images obtained from a moving camera at different view points after data association.

### 3.2. Data Association by Maximum Likelihood of Measurement

If the measurements only provide a landmark's position information, but the landmark's attribute information is empty, the method presented by Bar-Shalom et al. (Bar-Shalom & Fortmann, 1998) for the data association with innovation will be used here. Their method can be briefly described as follows:

Innovation is the value of difference between measurement  $Z(k)$  and predicted measurement  $\hat{Z}(k)$ , and expressed by  $\nu(k)$

$$\nu(k) = Z(k) - \hat{Z}(k). \quad (14)$$

In order to define a measurement validation region, the innovation needs to be normalized as follows:

$$\varepsilon_v(k) = v(k)^T S_v(k)^{-1} v(k) \quad (15)$$

where  $S_v(k)$  is the innovation covariance matrix, and it is defined as

$$S_v(k) = H(k)P_v(k)H(k)^T + R_v(k) \quad (16)$$

where  $P_v(k)$  is the state vector's estimated covariance matrix at step  $k$ , and  $\varepsilon_v(k)$  has a  $\chi^2$  distribution with  $n_z$  degrees of freedom ( $n_z$  is the dimension of the measurement  $Z$ ). The validation technique is based on this innovation. If a measurement is inside a fixed region of a  $\chi^2$  distribution, then this observation is accepted; otherwise the observation is rejected.

#### 4. Estimation of Robot Pose and Landmark Positions

In the SLAM problem, a robot pose and landmark positions at time  $k+1$  are unknown. They need to be estimated by using input information  $U$ , measurement information  $Z$ , and robot pose and feature position information, at time  $k$ . Stacking Equation (4) and Equation (5), the system process model can be expressed as

$$X_s(k+1) = \begin{bmatrix} X_v(k+1) \\ L_1(k+1) \\ L_2(k+1) \\ \vdots \\ L_m(k+1) \end{bmatrix} = \begin{bmatrix} X_v(k+1) \\ L(k+1) \end{bmatrix} = \begin{bmatrix} F(X_v(k), U(k), \mu(k)) + \omega(k) \\ L(k) \end{bmatrix} \quad (17)$$

and the measurement model is

$$Z(k) = H(X_v(k), L(k)) + \eta(k). \quad (18)$$

Both process model (Equation(17)) and measurement model (Equation(18)) are nonlinear equations. A straightforward method to solve this problem is the Extended Kalman Filter (EKF). Due to the high dimensions of the state vector and the need for linearization of the non-linear models, the EKF is not computationally attractive. The particle filter-based fast SLAM approach will be applied to solve this problem.

##### 4.1. Particle Filter

From the view point of probability, the estimation of a robot pose and landmark positions involves computing their posterior probability density function (PDF),  $p(X_v(k), L(k) | Z(k), X_v(k-1), U(k-1))$ , based on the prior probability density function  $p(Z(k) | X_v(k))$  and  $p(X_v(k-1) | Z(k-1), X_v(k-2), U(k-2))$ ,

where  $X_v(k)$  is the robot state,  $L(k)$  is the landmark state,  $Z(k)$  is the measurement,  $U(k)$  is the input to the system, at time  $k$ . This is the well-known Bayesian approach. According to the definition of a robot model and landmark model in the previous section 3.2, the PDF for a SLAM problem  $Bel(X_v(k+1))$  at time  $k+1$  can be defined as (Thrun & Burgard, 1998)

$$Bel(X_v(k+1)) = p(X_v(k+1) | Z(k+1), X_v(k), U(k)). \quad (19)$$

The solution for the SLAM problem is to estimate the maximum of the  $Bel(X_v(k+1))$ . Bayes' formula can be used on Equation (19) to simplify its implementation.

$$\begin{aligned} Bel(X_v(k+1)) &= \frac{p(Z(k+1) | X_v(k+1), X_v(k), U(k)) p(X_v(k+1) | X_v(k), U(k))}{p(Z(k+1) | X_v(k), U(k))} \\ &= \xi p(Z(k+1) | X_v(k+1), X_v(k), U(k)) p(X_v(k+1) | X_v(k), U(k)) \end{aligned} \quad (20)$$

where  $\xi$  is the value of the inverse denominator and is assumed to be a constant. It is known that the measurement  $Z(k+1)$  is only dependent on the current pose  $X_v(k+1)$  and is not influenced by previous pose  $X_v(k)$  and robot movement  $U(k)$ . Therefore, Equation (20) can be simplified into

$$Bel(X_v(k+1)) = \xi p(Z(k+1) | X_v(k+1), X_v(k), U(k)) p(X_v(k+1) | X_v(k), U(k)) \quad (21)$$

By applying the total probability theorem to the second item of the right hand side of Equation (21), then

$$\begin{aligned} Bel(X_v(k+1)) &= \xi p(Z(k+1) | X_v(k+1)) \\ &\int p(X_v(k+1) | X_v(k), U(k)) p(X_v(k) | Z(k), X_v(k-1), U(k-1)) dX_v(k) \end{aligned} \quad (22)$$

$$Bel(X_v(k+1)) = \xi p(Z(k+1) | X_v(k+1)) \int p(X_v(k+1) | X_v(k), U(k)) Bel(X_v(k)) dX_v(k) \quad (23)$$

where  $p(Z(k+1) | X_v(k+1))$  is the sensor observation model, which can be calculated from Equation (18);  $p(X_v(k+1) | X_v(k), U(k))$  is the system evolution model, which can be calculated from Equation (17). The integration in the Equation (23) is a difficult challenge to solve the SLAM problem efficiently; therefore, a new algorithm must be designed.

The Monte Carlo based particle filter can be used to overcome the implementation challenge in Equation (23). In the particle filter,  $Bel(X_v(k))$  is expressed as a set of particles and every particle is propagated in time according to the state process model such as Equation

(17). The weight of every particle is calculated based on the observation model from the Equation (18). The robot pose and landmark position can be computed from the sum of the weighted samples. The particles should be re-sampled for the next step's estimation. Implementation of a particle filter is summarized in Algorithm.1.

Input: Robot movement  $U(k)$ , sensor measurement  $Z(k+1)$  and sample number  $N$   
 Output: Robot pose and features position

- 1: initialize state with  $p(X_v(0))$
- 2: repeat
- 3: for every particle  $i$  do
- 4: assign distribution using  $p(X_v(k+1) | X_v(k), U(k))$
- 5: end for
- 6: for every particle  $i$  do
- 7: compute weight  $w$ , using  $p(Z(k+1) | X_v(k+1))$
- 8: end for
- 9: calculate robot pose & landmark position from particles & associated weight
- 10: re-sample the particles
- 11: until robot stop navigation

Alg. 1. Particle filter implementation for robot pose and feature position

Particle filtering can be used for any process and observation models. In the Kalman filter and the extended Kalman filter, the basic requirement is that the error of process model and observation model should be Gaussian distribution. In most cases, this requirement is too restrictive. The particle filter has been called bootstrap filter (Gordon, 1997), condensation (Isard & Blake, 1998), or Monte Carlo filter (Dellaert et al., 1999). In recent years, this method has been successfully used in problems of object tracking (Hue et al., 2002) and mobile robot localization (Dellaert et al., 1999) (Thrun et al., 2001).

#### 4.2. Fast SLAM

Fast SLAM is an approach to separate the SLAM problem into a robot pose and landmark position estimation that is conditioned on the robot pose. The term was first introduced in (Montemerlo et al., 2002). The implementation of FastSLAM is an example of the Rao-Blackwellised particle filter (Doucet et al., 2000) (Murphy, 1999).

Input: Robot movement  $U\{k\}$ , sensor measurement  $Z(k + 1)$ . and sample number  $N$

Output: Robot pose and detected features position

```

1: initialize state with  $p(X_v(0))$ 
2: repeat
3:   for every particle  $i$  do
4:     proposal distribution using  $p(X_v(k+1) | X_v(k), U(k))$ 
5:   end for
6:   obtain observations  $Z(k + 1)$ 
7:   data association for the observation data
8:   for every particle  $i$  do
9:     compute weight using  $p(Z(k+1) | X_v(k+1))$ 
10:  end for
11:  re-sample the particles
12:  if current observed feature exists in the map then
13:    for every particle  $i$  do
14:      for every observed feature do
15:        update the state of the robot
16:      end for
17:    end for
18:  end if
19:  if current observed feature is not in the map (new detected features) then
20:    for every particle  $i$  do
21:      add the new detected features to the map based on the robot pose and observation
22:      to the features
23:    end for
24:  end if
25: until robot stop navigation

```

Alg. 2. FastSLAM implementation

From the previous definition of a SLAM problem, the system state estimation could be written as

$$Bel(X_v(k + 1)) = p(X_v(k + 1), L(k + 1) | Z(k + 1), X_v(k), U(k)) \quad (24)$$

This expression can be factored into two parts according to [8].

$$Bel(X_v(k + 1)) = p(X_v(k + 1) | Z(k + 1), X_v(k), U(k)) \prod_{i=0}^m p(L_i(k + 1) | X_v(k + 1), Z(K + 1), U(k)) \quad (25)$$

The estimate expression is decomposed into  $m+1$  estimations. One of them is for the robot pose estimation, and  $m$  of them are for landmark estimation based on the estimated robot pose. The implementation of FastSLAM is summarized in Algorithm 2. In this thesis, this



algorithm is applied for the SLAM problem. The particle filter is implemented to calculate its related conditional densities for robot and landmarks.

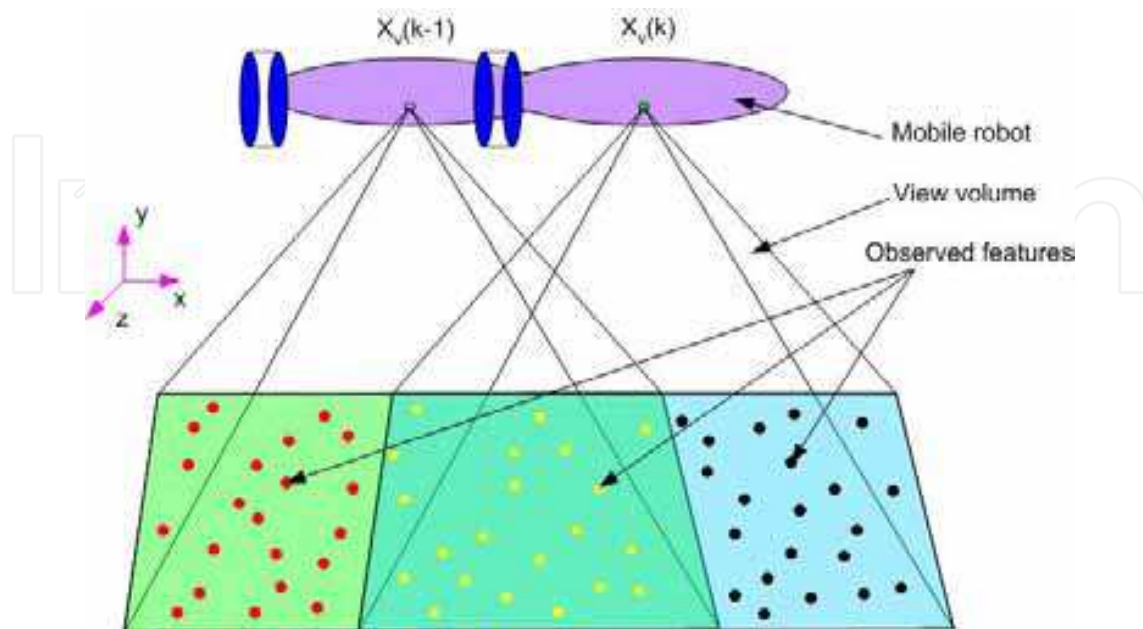


Fig. 6. SLAM implementation with features observed by a sensor.

The idea for FastSLAM can be obtained from Figure 6. We assume that all the observed landmarks by a sensor at robot position  $X_v(k-1)$  exist in a map. Then, the observed landmarks at robot position  $X_v(k)$  can be divided into two groups. Some of them are already in the map and are labeled as small yellow squares in Figure 6, which are called old landmarks; some of them are new landmarks and do not yet exist in the map and are expressed with small black dots in Figure 6. The measurements from the old landmarks are applied for the robot pose estimation at time  $k$ . The measurements from the new landmarks are used to estimate the new landmark positions in the global coordinate system based on the robot position. Then, all the new landmarks are added into the map.

In the fast SLAM approach, the factorizing assumption step turned the high dimension  $6 + 3 \cdot m$  of the SLAM problem into the low dimension (6 and 3) problem's combination, which greatly improves the computation efficiency, but it is assumed that the estimated robot pose has an accurate value. This assumption is not true and will cause errors in the step for the landmark position estimation in a global frame. This is the trade-off between efficiency and accuracy.

Another assumption in the fast SLAM is that the measurement of every detected landmark is independent of the other landmarks in the working area of the robot; therefore, the covariance between two landmarks will be zero. In other methods, such as the EKF or Particle filter, robot pose and all the detected features position are estimated in one state vector, and this may cause the covariance between two landmarks to be other than zero. In most of the cases, these values came from the algorithm design.



## 5. Multi-sensor Fusion for 3D SLAM

A mobile robot is usually equipped with many sensors which will work together. Fusing the measurements from more than one sensor will provide a more accurate estimation than by using only one sensor's measurement. An example of multi-sensor fusion is shown in Figure 7, where an underwater mobile robot is equipped with a stereo camera and communicated with a set of buoys on the surface.

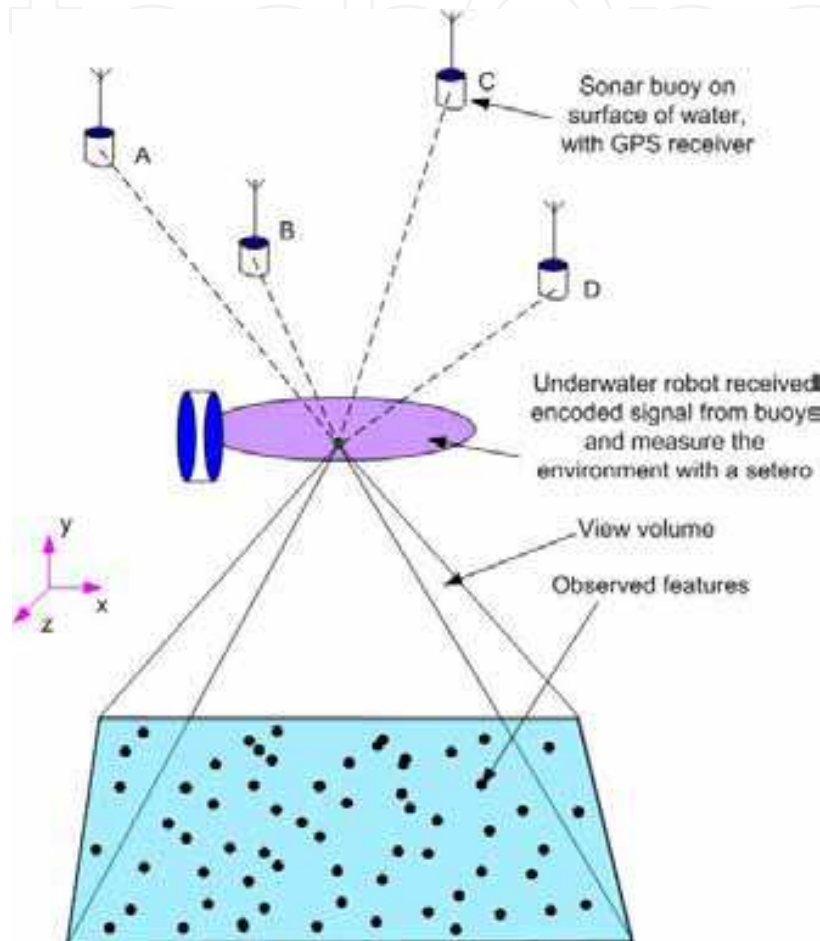


Fig. 7. A sensor fusion example for a stereo camera and a set of buoys.

This kind of system structure has two advantages for the underwater robot. The buoys can provide a long range of measurements, while the measurements can be applied to estimate the robot position in a global frame. The stereo camera can provide detailed information of the immediate environment, which will be used for SLAM in a local frame. Integration of both can solve the SLAM problem in a large area for the underwater robot. The buoy system has been proposed using acoustic sensor by (Liu & Milios, 2005).

### 5.1. Synchronization for Multi-sensor Fusion

All the sensors in a system cannot work at the same speed or frequency, such as in Figure 7. Synchronization for multi-sensor fusion is an important issue. Usually, measurement frequency for each sensor is different, and their measurement time will not coincide. An instance of the estimated robot position from stereo camera and buoys with time stamp is

shown in Figure 8. The buoys will provide a long range estimate in a global frame, which has less accumulated errors than that with a camera. Therefore, when the estimate from buoys at time  $t_{b,k}$  is obtained, the robot position from buoys at  $t_k$  should be estimated by interpolation as

$$X_b(t_k) = \xi_1 X_b(t_{b,k-2}) + \xi_2 X_b(t_{b,k-1}) + \xi_3 X_b(t_{b,k}) \quad (26)$$

where  $b$  means buoy and  $k$  means time stamp;  $X_b(t_{b,k-2})$ ,  $X_b(t_{b,k-1})$  and  $X_b(t_{b,k})$  are robot position estimated by buoys at time  $t_{b,k-2}$ ,  $t_{b,k-1}$  and  $t_{b,k}$ , respectively. And  $\xi_1$ ,  $\xi_2$  and  $\xi_3$  are coefficients related to the corresponding time stamps.

$$\xi_1 = \frac{(t_k - t_{b,k-1})(t_k - t_{b,k})}{(t_{b,k-2} - t_{b,k-1})(t_{b,k-2} - t_{b,k})} \quad (27)$$

$$\xi_2 = \frac{(t_k - t_{b,k-2})(t_k - t_{b,k})}{(t_{b,k-1} - t_{b,k-2})(t_{b,k-1} - t_{b,k})} \quad (28)$$

$$\xi_3 = \frac{(t_k - t_{b,k-2})(t_k - t_{b,k-1})}{(t_{b,k-2} - t_{b,k-2})(t_{b,k-2} - t_{b,k-1})} \quad (29)$$

The uncertainty  $P_b(t_k)$  of the position  $X_b(t_k)$  from buoys can be obtained from Equation (26) based on the fact that all the estimated robot positions are independent variables. The robot position is estimated by using the measurement between the current robot position and buoys, instead of previous robot position. This is

$$P_b(t_k) = \xi_1^2 P_b(t_{b,k-2}) + \xi_2^2 P_b(t_{b,k-1}) + \xi_3^2 P_b(t_{b,k}) \quad (30)$$

where  $P_b(t_{b,k-2})$ ,  $P_b(t_{b,k-1})$  and  $P_b(t_{b,k})$  are uncertainties of robot position estimated by buoys at time  $t_{b,k-2}$ ,  $t_{b,k-1}$  and  $t_{b,k}$ , respectively.

The estimated robot position from buoys at time  $t_k$  (see Figure 8.) can be obtained by using Equation (26) and Equation (30). At this time, fusing the robot estimation from the stereo camera and buoys is possible.

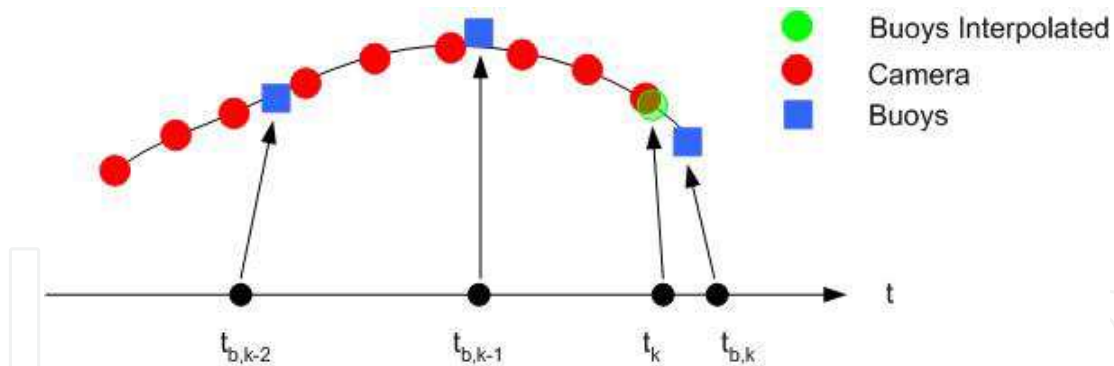


Fig. 8. A stereo camera and buoys estimation with time stamps.

It is very important to ensure that all the sensors installed in the same system have the same time signature. If there are more than two computers working for the same measurement system, it is necessary to calibrate them.

It must be pointed out that three points are used for the interpolation in this section. If there are more than three robot positions, using only three of them is applied for the interpolation; of course, higher order interpolation can be used. If is only two robot positions (points), linear interpolation could be applied.

## 5.2. General Sensor Fusion Mechanism

All the sensors in the system are applied for one purpose: reliable and accurate robot pose and map estimation. The sensors will not work properly all the time; therefore, a complementary fusion mechanism is designed in this thesis. Figure 9. is a sensor fusion architecture for a stereo camera and a set of buoys.

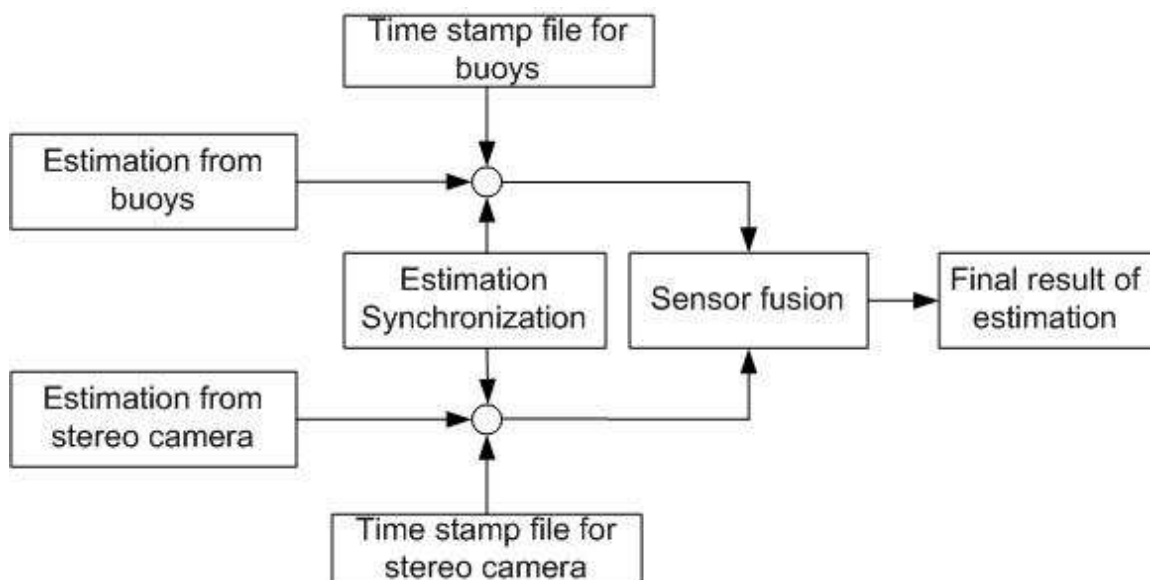


Fig. 9. Sensor fusion architecture for camera and buoys.

Before sensor fusion is applied, each sensor in the system estimates the robot pose and/or builds a map independently. When the synchronized estimate is obtained, sensor fusion will be performed. Assuming that the error from each sensor follows the Gaussian distribution with zero mean and known covariance, from [1], the fusion is carried out by

$$X_v(t_k) = \frac{P_b(t_k)}{P_b(t_k) + P_c(t_k)} X_c(t_k) + \frac{P_c(t_k)}{P_b(t_k) + P_c(t_k)} X_b(t_k) \quad (31)$$

$$P_v(t_k) = \frac{P_b(t_k)P_c(t_k)}{P_b(t_k) + P_c(t_k)} \quad (32)$$

where  $X_c(t_k)$  and  $X_b(t_k)$  are the estimated robot pose at time  $t_k$  by the stereo camera and buoys, and  $P_b(t_k)$  and  $P_c(t_k)$  are their associated covariance, respectively.

It must be pointed out that the relative drift of the sensor estimation has a big influence to the sensor fusion result. Ideally, both of the sensor estimation should be overlapped in their estimation uncertainty's boundary. If both of the sensor estimation did not overlap in their uncertainty boundary, the fusion result will not be reliable. To avoid big relative drift of the sensor fusion, we used multi-map mechanism, which means that a new estimation will start as soon as the estimation uncertainty reaches a threshold.

By using this sensor fusion mechanism and system designed for an underwater mobile robot, it is possible to estimate a globally-consistent robot position in a large working area. The following section discusses globally-consistent map building.

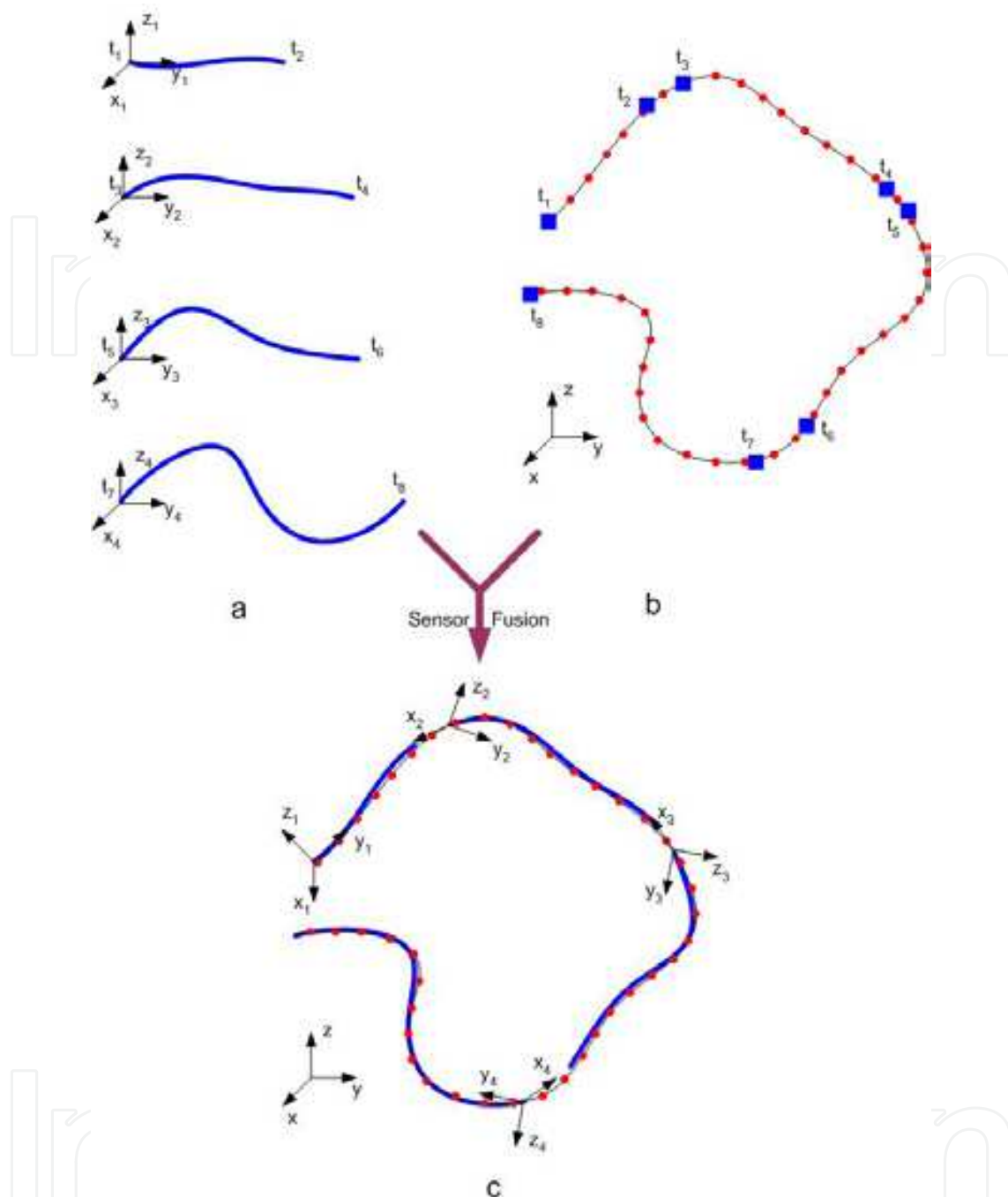


Fig. 10. 3D SLAM by fusion estimation from camera and buoys. (a) Local robot position and map (for simplicity, map is not displayed in this figure) from camera with time stamps. (b) Robot position estimated from buoys with time stamps. (c) Final results after fusing the information from both camera and buoys.

## 6. Globally-Consistent 3D SLAM for Mobile Robot in Application

The basic requirement for SLAM using stereo vision is that every two consecutive images have to provide enough overlapped features in all the images. In real applications, this requirement may be too strict. If this requirement can not be satisfied, the SLAM process

based on image information will be stopped. In this case, even though the robot has another sensor installed (such as a GPS or buoys), it can not obtain a globally-consistent 3D path and a globally-consistent 3D map; therefore, it is necessary to design a new algorithm to solve the 3D SLAM problem in real application.

Assuming that two sensors (stereo camera and buoys or GPS) will be applied, both of which work separately, where the buoys will estimate a globally-consistent robot 3D path and the stereo camera will estimate a set of local maps and the robot's 3D paths in a local coordinate system. This means that when the two consecutive images provide enough overlapped features, the SLAM algorithm begins to work based on the local frames. The estimate results are also stamped with global time. The scenario of this processing is shown in Figure 10. There are four local robot path and associated maps (not shown) based on local frame which is estimated by the stereo camera's measurement with the algorithm of SLAM in Figure 10 (a). For simplicity, the maps are not drawn. These estimated path fragments are time stamped, which will be applied for sensor fusion. For the estimate by buoys measurement in Figure 10(b), the red dots are the robot's position at each time, and the blue squares are the time markers which are corresponding to the time stamps in Figure 10. (a). The final globally-consistent robot path and map are expressed in Figure 10. (c). There are several different steps required to obtain these: (1) robot path parameterizing; (2) robot position association; (3) transformation estimate from local coordinate system to global coordinate system; (4) globally-consistent map integration, and; (5) globally-consistent robot position.

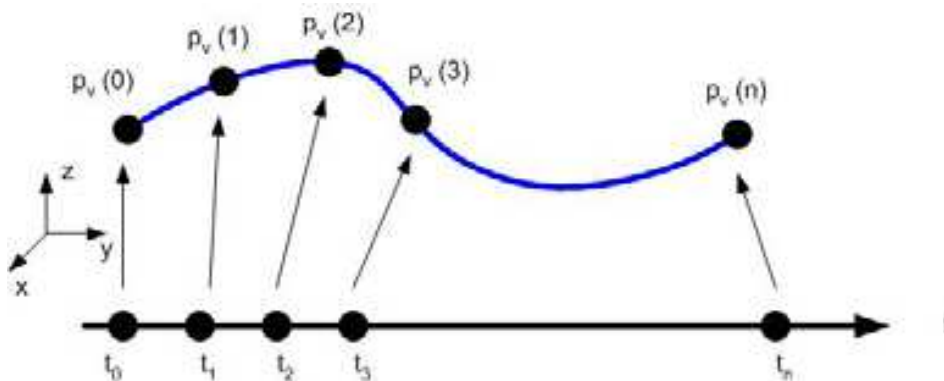


Fig. 11. Robot path parameterizing with B-spline.

### 6.1. Robot Path Parameterizing

For 3D SLAM, connecting the robot position over time will form a curve in 3D space. A widely-used method to construct a 3D curve to smoothly pass all the discrete position points is the B-spline interpolation (Piegl & Tiller, 1997). Assume that a sequence of robot 3D positions,  $p_v(k)$ , and its related covariance  $P_{vv}(k)$  ( $k = 0, \dots, n$ ), are obtained from a local coordinate system, and its corresponding time is  $t_k$  (Figure 11.). A parametric B-spline of degree 3 to pass these points is defined as

$$S(u) = \sum_{i=0}^n p_v(i) N_{i,3}(u) \quad u \in [0,1] \quad (33)$$



where  $N_{i,3}(u)$  are the B-spline basis functions of degree 3, defined with respect to the knot vectors  $u = \{u_0, u_1, \dots, u_{n+3+1}\}$ , with  $u_0 = u_1 = u_2 = u_3 = 0$  and  $u_{n+1} = u_{n+2} = u_{n+3} = u_{n+3+1} = 1$ , and its parameter  $u$  at any time can be obtained by

$$u = \frac{t - t_0}{t_n - t_0} \quad t \in [t_0, t_n] \quad (34)$$

A B-spline base function  $N_{i,p}(u)$  of degree  $p$  can be calculated by

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{any others} \end{cases} \quad (35)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (36)$$

$$\text{define } \frac{0}{0} = 0.$$

By using the B-spline function in Equation (33), for any time  $t$  between time  $t_0$  and  $t_n$ , it is easy to obtain the position of the robot. Its related covariance can be obtained by

$$P_{vv}(u) = \sum_{i=0}^n p_v(i) (N_{i,3}(u))^2 \quad u \in [0,1]. \quad (37)$$

## 6.2. Robot Position Association

From the buoys, the robot position is estimated during its navigation. The positions,  $X_b(k)$  ( $k = 0, \dots, m$ ), are denoted by small black squares on the red curve in Figure 12, and their corresponding times are shown on the time axis with  $tb_k$  ( $k = 0, \dots, m$ ). The robot path estimated from the camera is expressed by the blue curve, which is constructed in the previous subsection. With the time information, the corresponding robot position,  $X'_v(k)$ , based on the local coordinate system ( $x' y' z'$ ), can be calculated from the Equation(33) and Equation(37).

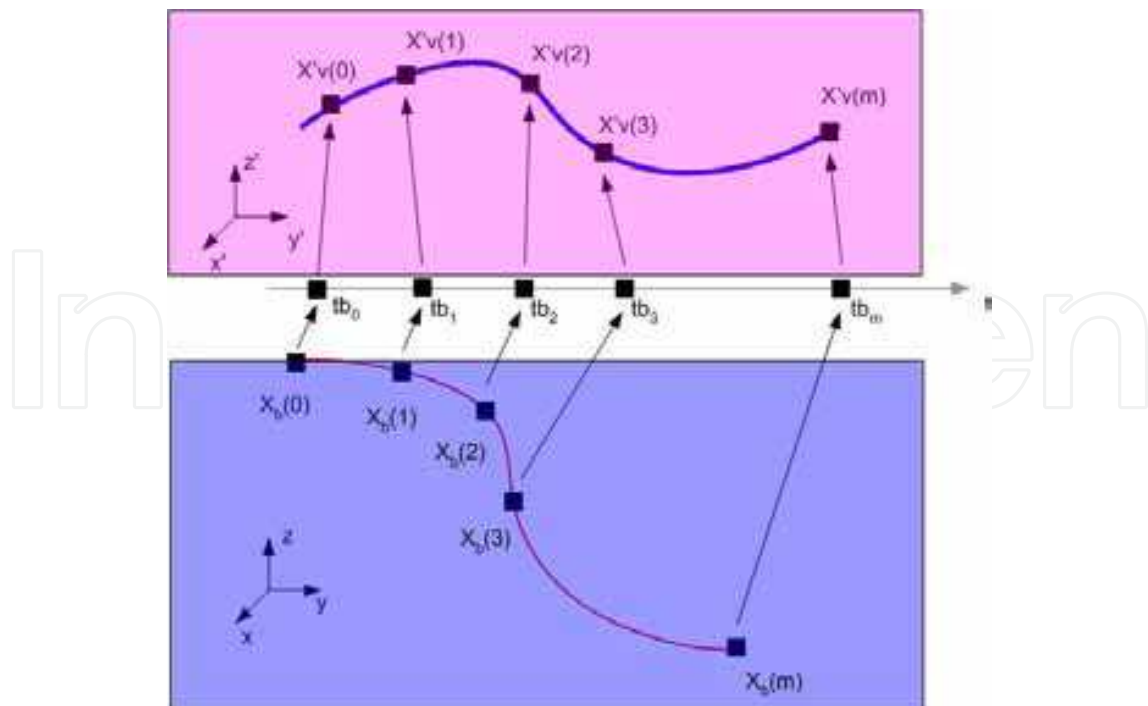


Fig. 12. Robot position association. The top is in local coordinate system, and the bottom is in global coordinate system.

### 6.3. Transformation between Frames

So far, two sets of corresponding 3D points,  $X_b(k)$  and  $X'_v(k)$  ( $k = 0, \dots, m$ ), are obtained (Note: we only use  $X'_v(k)$  position information here since  $X_b(k)$  only contain position information). If the number of corresponding points,  $m$ , is greater than 3 and all the points in each coordinate system are not in a straight line, it is then possible to estimate the transformation (rotation and translation) between these two data sets. If all the estimated 3D points in one of its local coordinate system are located on a line or are very close to a line, it is impossible to estimate the orientation of this transformation. In this situation, other information such as a robot direction from a digital compass will be needed.

### 6.4. Globally-Consistent Map Integration

Assuming that the transformation for a local map  $k$  ( $k = 1, \dots, n$ ) from a local coordinate system to a global coordinate system is expressed by  $R_k$  and  $T_k$ , for rotation and translation, respectively, a map established in the local coordinate system can be then transformed to the global coordinate system by

$$L_k^g(i) = R_k L_k^l(i) + T_k \quad (38)$$

where  $L_k^g(i)$  is the landmark  $i$ 's position in the global coordinate system, and  $L_k^l(i)$  is the landmark  $i$ 's position in the local coordinate system ( $i = 1, \dots, M$ ).

```

Input: a sequence of images  $I_k, k=1, \dots, N$ , measurements from buoys, and a file of time
stamps for each sensor
Output: a globally consistent map  $G$  and robot path  $p_v$ 

1:  k=2, numMap=0, flag=new Map_start, Segment=[ ]
2:  repeat
3:    robot path estimated from the measurements of buoys
4:    extract SIFT features from image  $I_k, I_{k-1}$ 
5:    if matched features between image  $I_{k-1}$  and  $I_k$  are more than  $m_l$  or the estimated
    position error is less than a certain threshold then
6:      if flag=new Map_start then
7:        numMap=numMap+1;
8:      end if
9:      SLAM estimation based on stereo camera's measurements
10:     Segment(k)=numMap
11:     flag=map_continue
12:   else
13:     flag=new Map_start
14:   end if
15: until robot stop navigation
16: robot path 3D curve from buoys estimation by B-spline
17: for each Segment do
18:   find corresponding point on 3D path curve
19:   estimate transformation parameters between local frame and global frame
20:   global map and path calculation
21:   integrate 3D map and path
22: end for

```

Alg. 3. Globally Consistent 3D SLAM based on Sensor Fusion

If there are several different local maps established on one working area, it will be necessary to transform all of them from each local coordinate system to a global coordinate system, and then integrate all of them to form a globally-consistent map.

### 6.5. Globally-Consistent Robot Position

Due to the bandwidth limitation of the communication or the nature of the sensing, the measurement frequency of the buoys is lower than the stereo camera's frequency. In order to obtain a continuously consistent 3D path, the robot path in a local coordinate system should be transformed to the global coordinate system according to the same method from the previous map transformation.

The algorithm to establish a globally-consistent 3D map is based on the sensor fusion shown in Algorithm 3. An overview of this algorithm is shown in Figure 13. The centre of this diagram belongs to Alg. 3, and the left part for local SLAM is obtained by a camera with the previous FastSLAM, and the global robot position is obtained with the EKF algorithms by buoys and GPS.

In the processing of local SLAM from a camera, there are two issues which need to be considered: estimation error and efficient map. In order to control the estimation error in the

local SLAM result, we always check the estimation error for robot path. If the estimated path error grows above a certain threshold, the current local SLAM processing is also stopped, in which case a new local SLAM processing will be started. For efficient map, we will discuss it in the next chapter.

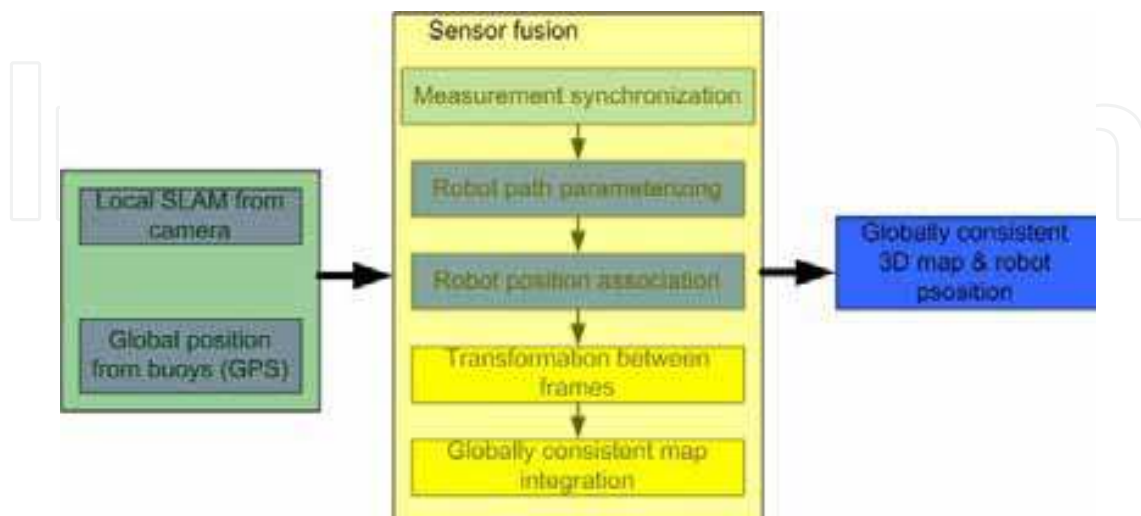


Fig. 13. Diagram for the globally-consistent 3D SLAM.

## 7. Summary

This chapter established an approach to solve the full 3D SLAM problem, applied to an underwater environment. First, a general approach to the 3D SLAM problem was presented, which included the models in 3D case, data association and estimation algorithm. For an underwater mobile robot, a new measurement system was designed for large area's globally-consistent SLAM: buoys for long-range estimation, and camera for short-range estimation and map building. Globally-consistent results could be obtained by a complementary sensor fusion mechanism.

By carefully investigating all the algorithms used for SLAM, we designed a new sensor fusion algorithm for large area SLAM, which will be suitable for most of application environment. Two types of sensors are needed for this algorithm: stereo camera (or laser scanner, radar) for local SLAM, and a sensor for robot path in the global coordinate system. Both the local and global paths were expressed by B-spline, and their corresponding 3D points of the associate robot path could be extracted. Transformation (translation and rotation) values from local coordinate system to global coordinate system could be estimated from these matched points, and, the local maps could be integrated to the globally-coordinate system, and a globally-consistent map and path could be obtained.

## 8. Reference

- Bar-Shalom, Y. & Fortmann, T. E. (1998). Tracking and Data Association. Academic Press  
 Dellaert, F.; Foxt, D.; W. Burgard, & Thrun, S. (1999). Monte Carlo Localization for Mobile Robots. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation, pages 1322-1328, Michigan

- Doucet, A., Godsill, S. & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197-208
- Gordon, N. J. (1997). A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33:353-358
- Hue, C.; Cadre, J. L. & Perez, P. (2002). Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38:791-812
- Isard, M. & Blake, A. (1998). Condensation-conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29:5-28
- Liu, H. & Milios, E. (2005). Acoustic positioning using multiple microphone arrays. *Journal of the Acoustical Society of America*, 117(5):2772-2782
- Montemerlo, M.; Thrun, S.; Koller, D. & Wegbreit, B. (2002). Fast SLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada
- Murphy, K. (1999). Bayesian map learning in dynamic environments. In *Proceedings (?) Neural Information Processing Systems (NIPS)*
- Piegl, L. A. & Tiller, W. (1997). *The NURBS Book*. Springer
- Se, S.; Lowe, D. & Little, J. (2003). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Int. Journal of Robotics Research*, 21:735-758
- Sonka, M.; Hlavac, V. & Boyle, R. (1999). *Image Processing, Analysis, and Machine Vision*. PWS Publishing
- Thrun, S. & Burgard, W. (1998). A probabilistic approach to concurrent mapping and localization for mobile robot. *Machine Learning*, 31:29-53
- Thrun, S.; Fox, D.; Burgard, W. & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128:99-141, 2001

IntechOpen



## **Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions**

Edited by XiaoQiChen

ISBN 978-953-307-001-8

Hard cover, 335 pages

**Publisher** InTech

**Published online** 01, May, 2009

**Published in print edition** May, 2009

Since the introduction of the first industrial robot Unimate in a General Motors automobile factory in New Jersey in 1961, robots have gained stronger and stronger foothold in the industry. In the meantime, robotics research has been expanding from fix based robots to mobile robots at a stunning pace. There have been significant milestones that are worth noting in recent decades. Examples are the octopus-like Tentacle Arm developed by Marvin Minsky in 1968, the Stanford Cart crossing a chair-filled room without human assistance in 1979, and most recently, humanoid robots developed by Honda. Despite rapid technological developments and extensive research efforts in mobility, perception, navigation and control, mobile robots still fare badly in comparison with human abilities. For example, in physical interactions with subjects and objects in an operational environment, a human being can easily relies on his/her intuitively force-based servoing to accomplish contact tasks, handling and processing materials and interacting with people safely and precisely. The intuitiveness, learning ability and contextual knowledge, which are natural part of human instincts, are hard to come by for robots. The above observations simply highlight the monumental works and challenges ahead when researchers aspire to turn mobile robots to greater benefits to humankinds. This book is by no means to address all the issues associated mobile robots, but reports current states of some challenging research projects in mobile robotics ranging from land, humanoid, underwater, aerial robots, to rehabilitation.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Peter Zhang, Evangelous Millos and Jason Gu (2009). General Concept of 3D SLAM, Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions, XiaoQiChen (Ed.), ISBN: 978-953-307-001-8, InTech, Available from: <http://www.intechopen.com/books/mobile-robots-state-of-the-art-in-land-sea-air-and-collaborative-missions/general-concept-of-3d-slam>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

[www.intechopen.com](http://www.intechopen.com)



[www.intechopen.com](http://www.intechopen.com)

IntechOpen

IntechOpen

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen