Absolute Localization Of Cycab

Gerald Chong Chin Hui

INDUSTRIAL ATTACHMENT TECHNICAL REPORT INRIA RHONE-ALPES

Dated	: 15/04/2005
Student	: Chong Chin Hui
Supervisor	: Christopher Tay Meng Keat
	Christophe Braillon
Team	: E-motion headed by Christian Laugier
Tutor	: Michel B Pasquier (Assoc Prof)

<u>Training Records</u>

<u>Introduction:</u>

The Cycab robot has no prior information about its environment; hence it has to build its own map. The building of map is done in parallel with estimating its position, therefore there is a need to counter the problem of Simultaneous Localization And Mapping (SLAM) [1]. When moving the Cycab, it is necessary to equip the Cycab with some localization ability. Therefore, the Cycab is equipped with a 2D sensor (SICK), and some artificial landmarks so that it can compute its position error with respect to the nominal path and correct it through a closed loop.

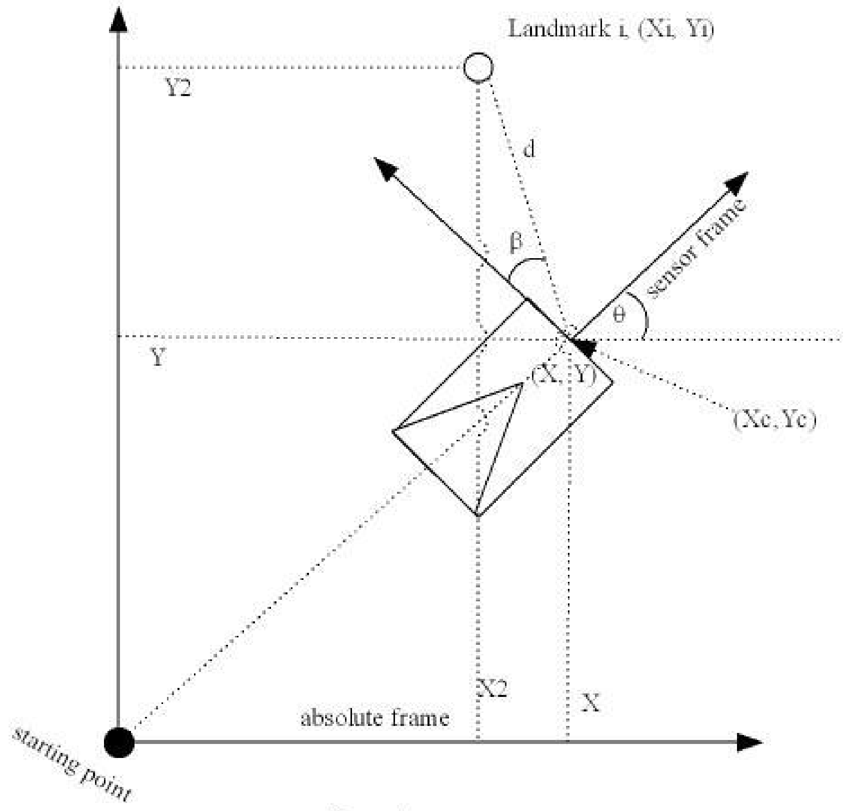


figure 1

The Cycab's position will be described by the position (X, Y, θ) reference to its starting position. All points that are with referenced to the Cycab's starting position will be known as absolute coordinates in the absolute frame. All observed landmarks position will be described by the position (Xi, Yi, θi). These coordinates are with referenced to the current Cycab's position and are known as the sensor frame coordinates.

Building of Map:

The Cycab will starts from stationary position and marked that position as (0, 0, 0). This is the starting position which is also the origin of the absolute frame. The database which stored the verify landmarks is known as the currentMap and it will be empty initially. When the Cycab moves, the odometry[1] data are tracked and used to estimate its current position. When some landmarks are observed, the sensor will provided the distances of these landmarks and their orientation with respect to the Cycab's current position. This information plus the estimated position from the odometry data will be able to provide the coordinates of the observed landmarks in absolute frame which will be stored in the database waitingMap. When the Cycab see a particular landmark for a defined number of times, it will move that landmark from the waitingMap to the currentMap. This landmark will then be known as verified landmark. Refer to figure 1, the absolute position (X1, Y1) of the landmark i with sensor frame coordinate of (Xi, Yi) is given by:

$$X2 = X + \frac{d}{\sin[\beta + 90 - \theta]} \quad \text{and,} \quad Y2 = Y + \frac{d}{\sin[90 - \beta]}$$

where (X, Y) is the position of Cycab in absolute frame estimated by the odometry data. The distance between two landmarks will be known as segment. Every 3 landmarks will be able to form a triangle. Triangle properties will refer to the 3 landmarks involved, segments of the 3 sides and the area of the triangle. Concurrently, while a landmarks is being observed and stored in waitingMap waiting for verification, all the possible segments and triangles that it can formed with the existing landmarks in waitingMap will be computed and stored in the database known as WDMap and WAMap respectively. When the landmark is verified and moved to the currentMap, all possible segments and triangle will be moved and stored in DMap and AMap respectively. Hence, DMap will contain all possible segment that can be formed by the verify landmarks while AMap will contain all the possible triangle properties.

1) odometry data provides the distance moved in the X and Y axis and the angle of Cycab's rotation

Localization and updating of Map:

DMap and AMap are vital for both localization and updating of map. These two processes are inter-dependent on each other. Since the artificial landmarks used are indifferentiable, the identification of the landmarks must make use of a method robust to the errors in estimation of the Cycab's position and the error due to erroneous detection. From figure 1, the matching of the sensor frame onto the absolute frame will only involved one rotation and one translation. A triangle is a rigid shape and its properties are invariant to these 2 transformations hence, AMap is needed to store all possible triangles combination for localization. If less than 2 landmarks are observed, the segments can be used instead. The detail of how two triangles will be matched will be cover later on.

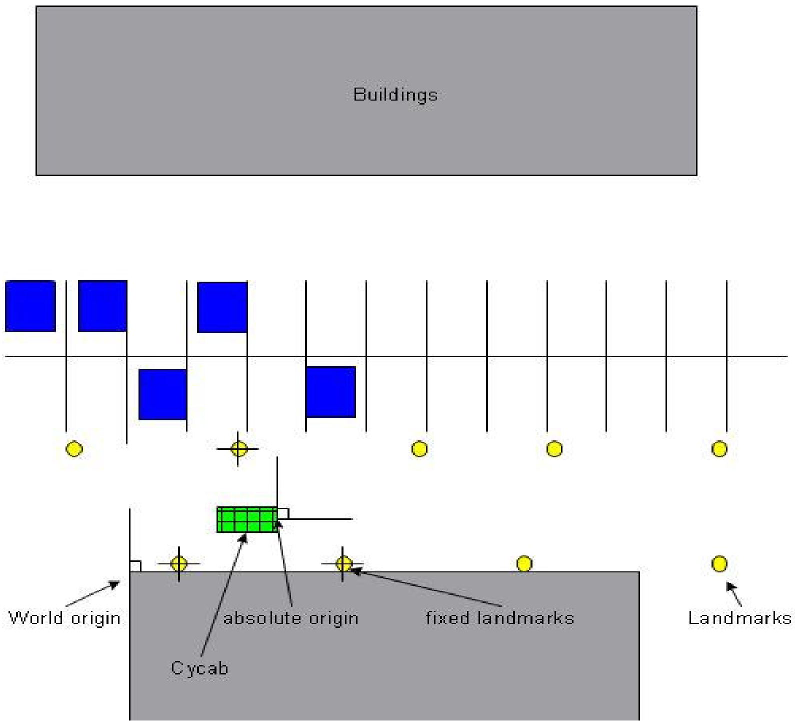
During localization, a set of landmarks observed will be known as the impacts points whose coordinates are in sensor frame. This set of impact points will have their possible triangles properties computed and a brute force search is done to find a similar triangle in the AMap. If no similar triangle is found, these set of impacts will be cast as newly discovered landmarks and stored in the waitingMap. If a match is found, it is first assumed that the 3 impact points involved are actually the 3 verify landmarks that formed the matched triangle in the AMap. Hence, from the triangle in AMap, we can have the absolute coordinates of the 3 impact points. Using these absolute values and the distances and orientations detected by the sensor, the Cycab's position (Xc, Yc) can be computed. Refer to figure 1:

$Xc = X2 + d[\cos(\beta + 90 - \theta)]$ And $Yc = Y2 - d[\sin(\beta + 90 - \theta)]$

The computed position (Xc, Yc) will be used to compare with the position (X, Y) estimated using the odometry data. If the 2 position are similar, the Cycab's has found its actual location otherwise, that matched triangle is the wrong triangle that might have the same properties. Concurrently, the Cycab will be doing updating of its map. Those impacts point deem as newly discovered landmarks will go through the process where their segments and triangles properties are computed and stored in the WDMap and WAMap respectively. Then they will be insert into the waitingMap until they are being verify and moved to the currentMap or deleted permanently. Map building provides AMap data for identification of impact points by localization while localization provides Cycab's position to assist in map building.

Problem Statement:

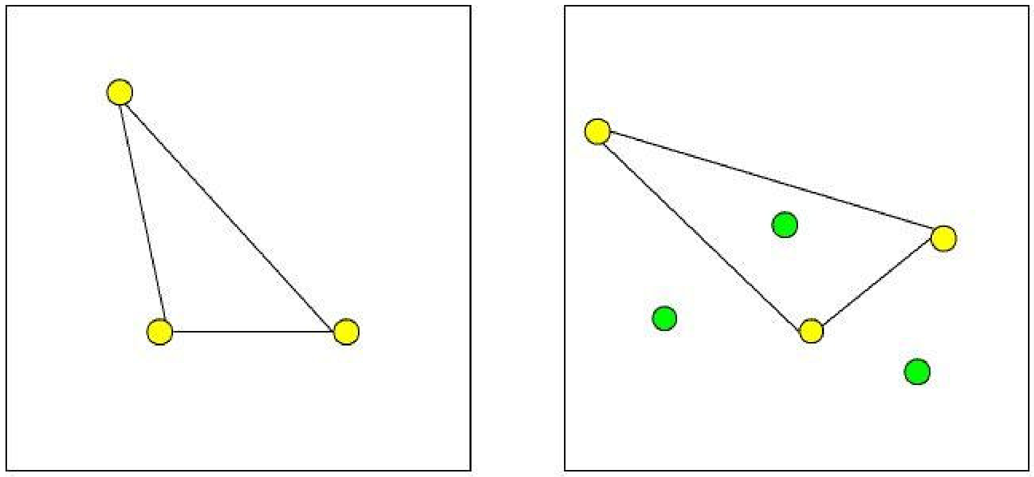
Refer to the figure of a car park below; There is a world origin, this origin is manually fixed and can be anywhere, in this case, the corner of a building is manually fixed as the world origin. Any coordinates with reference to this origin will be named as world coordinates and the frame they are in as the world frame. ***The objective is to allow the Cycab to localize itself reference to this world frame which is independent of the starting position of the Cycab, the absolute frame*** mention earlier on. The process to localization the Cycab in this world frame will be known as *World Localization*.



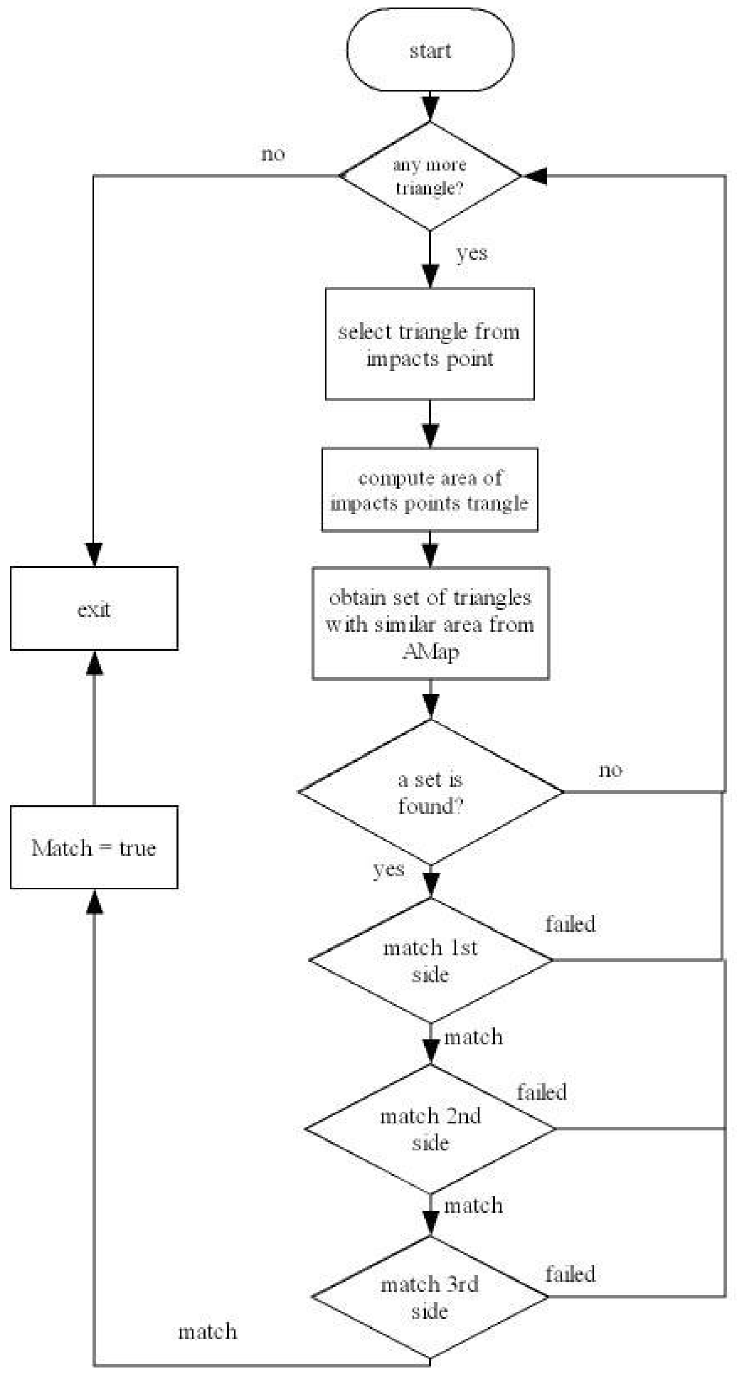
Requirement Analysis:

Without prior knowledge of the environment, the Cycab will perform SLAM to build the map as well as estimating its location with respect to the starting point. A minimum of 3 landmarks will be declared as fixed known landmarks. The world coordinates of these fixed known landmarks will be manually measured with reference to the world origin and feed to the Cycab's program. The main task is to locate and identify these known fixed landmarks during the Cycab's motion. Once the Cycab detected and verified these fixed known landmarks, mapping can be perform to obtain the transformation matrix needed to convert any absolute coordinates to world coordinates.

<u>Method of Matching:</u>


3 observed Impacts point in sensor frame
Verified landmarks in AMap

While the Cycab is in motion, the currentMap, DMap and AMap are being built and updated. At each instant where a set of impacts point are detected, SLAM will perform matching of the impacts point with those landmark in AMap. As mention before, triangle is chosen because its angles and length of sides are invariant to transformation. A brute force search will be performed on the AMap. The purpose of the search is to find from the AMap, a similar triangle to that of the impact points. The figure above shows the triangle formed by the impact points on the left and a similar triangle found in the AMap at a particular instant. According to the triangle SSS theorem [2], two triangles with all 3 sides equal are known as similar triangle. Two triangle that are similar will no doubts have the same area, hence to minimize the search, area of triangle will be used to filter and obtain a sets of triangle with the same area. This set of triangles will then have their three sides compared to find the one which will satisfy the SSS theorem. In the case, when lesser than three impacts point are detected, the segment will be used instead. If only one impact point is detected, no matching will be performed and that impacts point will be placed into waitingMap for verification. During world localization, all matching that are needed to be performed will use the same method of triangle matching. The following flow chart will explained the method of matching using triangle:



<u>Assumption made:</u>

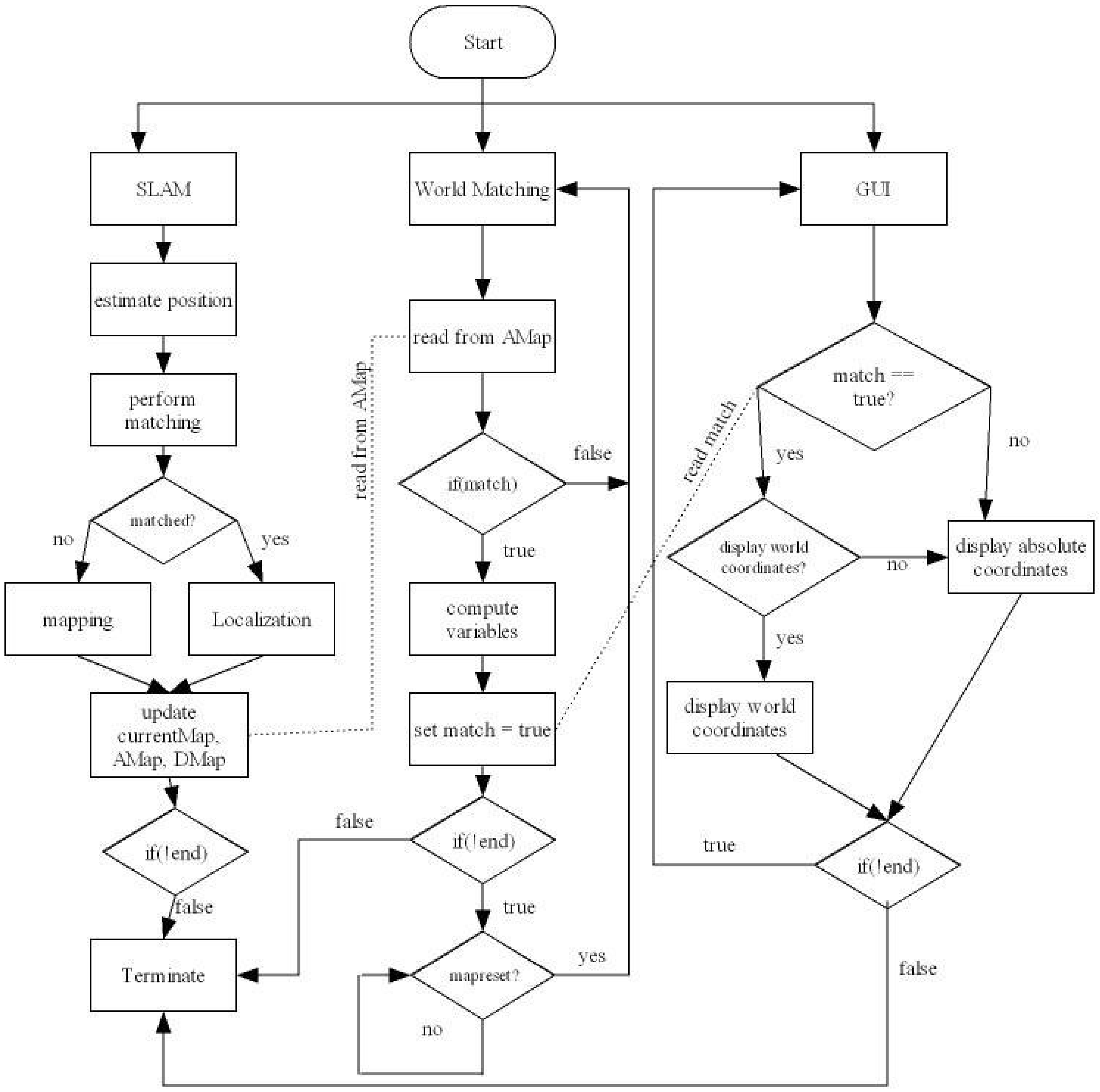
Two assumptions had to be made to ensure proper performance by the world localization:

1) The position and properties of the fixed known landmarks are unique.
2) The Cycab will eventually see those fixed known landmarks.

Process threading:

There will be a total of 3 main threads running simultaneously

1) SLAM thread
    a. perform both localization and mapping concurrently
    b. updating of waitingMap, currentMap, WDMap, WAMap, DMap, AMap
2) World localization thread
    a. Perform triangle matching between fixed known landmarks and AMap
    b. Compute transformation matrix to map absolute frame to world frame
    c. Compute world coordinates from absolute coordinates
3) GUI interface thread
    a. to represent the Cycab's location in world frame
    b. provided user the choice to display either world or absolute coordinates



Functional Analysis:

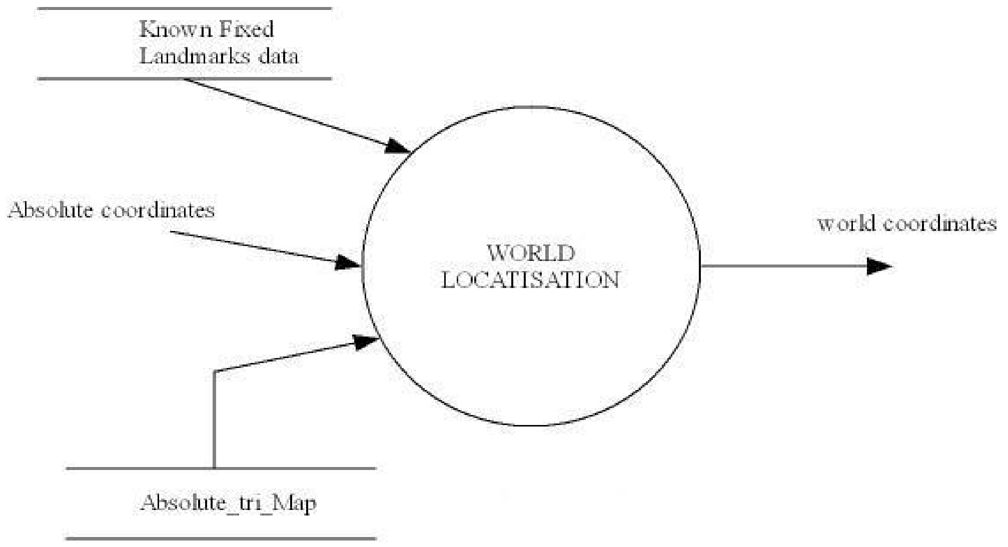


Figure1) 1st level FD

In order to perform world localization, we required 3 inputs.

1) Known fixed landmarks data

Known fixed landmarks data will contain the (X,Y) coordinates of a minimum of 3 fixed known landmarks manually obtain with reference to the chosen world origin. All the possible combination of triangles, their area, the length of their sides and the landmarks involved will be included as well. Therefore, for N number of fixed known landmarks, the number of triangles combination will be:

$$\frac{N(N\text{-}1)(N\text{-}2)}{6} \qquad ,N > 3 \text{ or } N = 3$$

2) Absolute triangle map (AMap)

The database contains the triangle properties of all possible triangles formed by the verified landmarks. These landmarks are in the Absolute frame. The AMap is built and updated by the SLAM thread.

3) Absolute coordinate

This coordinate is the Cycab's position in the absolute frame estimated by the SLAM thread. They will be fed into world localization thread to be map onto the world frame.

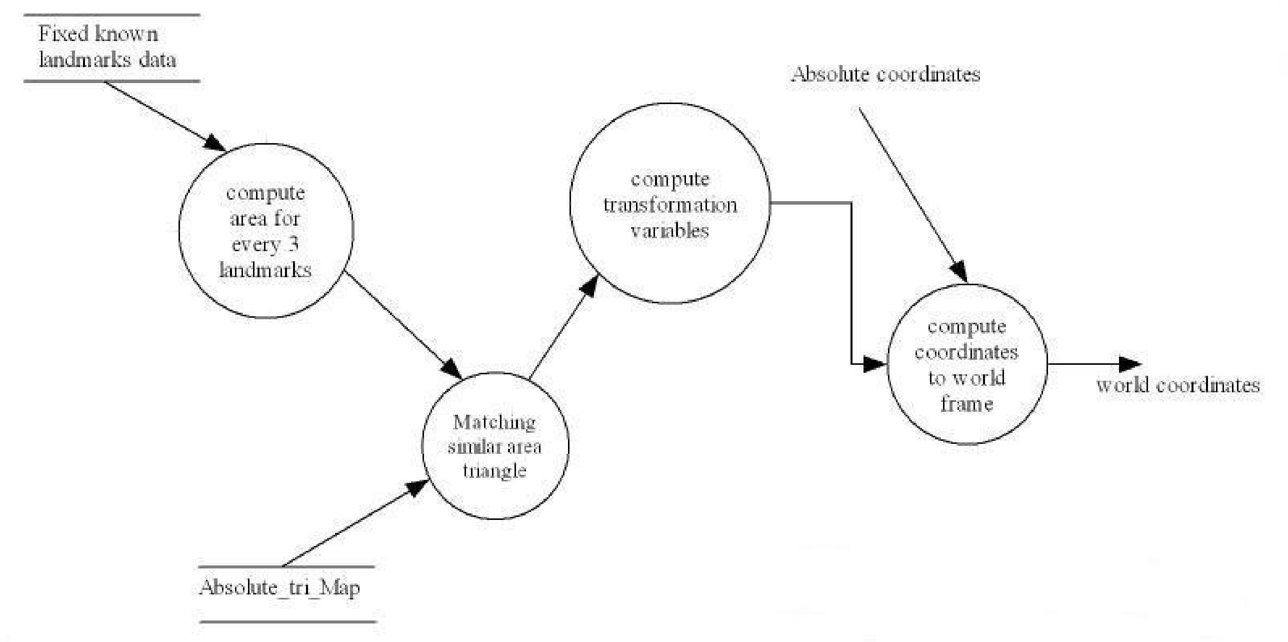

Fixed known landmarks data
compute area for every 3 landmarks
Matching similar area triangle
Absolute_tri_Map
compute transformation variables
Absolute coordinates
compute coordinates to world frame
world coordinates

Figure 2) 2nd level FD

The world localization thread will run concurrently with the SLAM thread. While AMap is empty, world localization will idle. Once the AMap is updated, world localization thread will perform matching of triangles from the fixed known landmarks database with those in the AMap. The method of matching is the same as the method used by the SLAM. Instead of matching impact points to AMap, known landmarks are used for matching. The thread will perform matching until a match is found. Since it is assume that position and properties of the fixed known landmarks are unique, there will only be one case of similar triangle match. Therefore, we can be sure that this will be the correct match. From the matched triangle of the AMap, the three verified landmarks absolute coordinates can be obtain.

A successful match will provides the following:

1) The three (X_i', Y_i') coordinates in the absolute frame which is the match of

2) The three (X_i, Y_i) coordinates in the world frame

Mathematic calculation:

Let (X_i',Y_i') be coordinate of landmarks in absolute frame

(X_i, Y_i) be coordinate of landmarks in world frame

(m, n) be coordinates of absolute origin in world frame

θ be orientation of absolute frame in respect to world frame

Transformation matrix will be:

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -m \\ 0 & 1 & -n \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Simultaneous Equation Result:

$$\cos\Theta = \frac{(X1'\text{-}X2')(X1\text{-}X2) - (Y1'\text{-}Y2')(Y2\text{-}Y1)}{(Y2 - Y1)^2 - (X2\text{-}X1)(X1\text{-}X2)}$$

$$\sin\Theta = \frac{(X1'\text{-}X2')(Y1\text{-}Y2) - (Y1'\text{-}Y2')(X1\text{-}X2)}{(X2\text{-}X1)(X1\text{-}X2) - (Y1 - Y2)^2}$$

$$m = X1 - Y1'sin\Theta - X1'cos\Theta$$

$$n = X1'sin\Theta + Y1 - Y1'cos\Theta$$

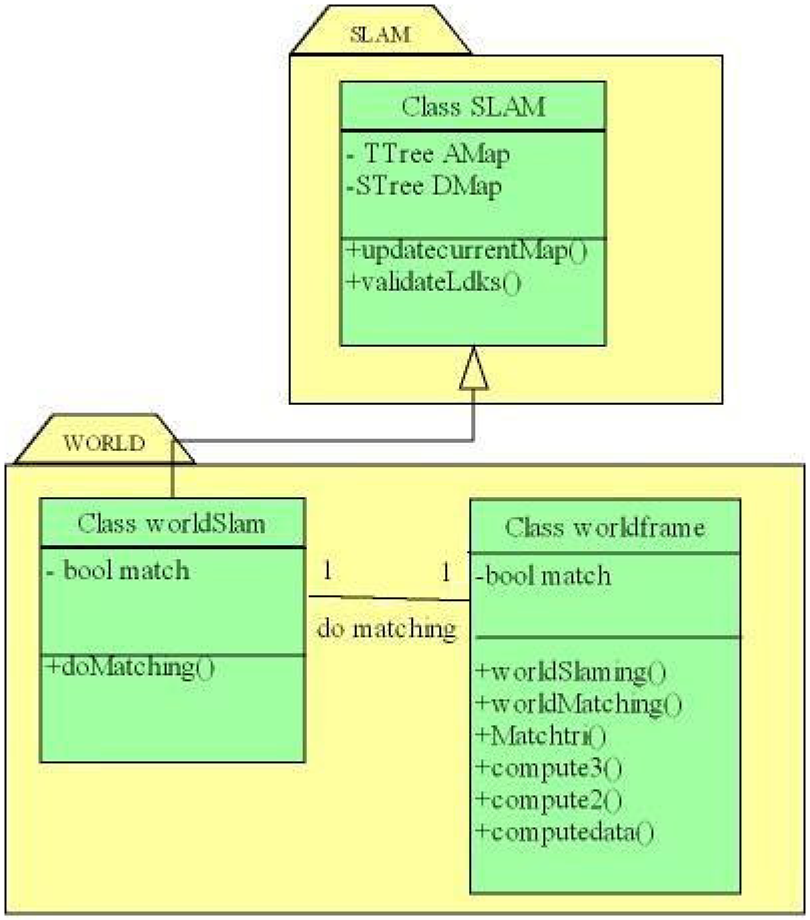
From the transformation matrix:

$$X = Y'sin\Theta + m + X'cos\Theta$$

$$Y = n + Y'cos\Theta - X'sin\Theta$$

Implementation:

The world localization module will be built on the existing Cycab's simulator program. Therefore the programming language C++ is used. The figure below shows the class diagram:



Class worldSlam is the subclass of Class SLAM. This will allowed the use of the variable AMap which is the absolute triangle map. All mathematical computation will be performed but the function from Class worldframe.

worldSlaming( )	➔ building triangles combination and queue them for matching.
worldMatching( )	➔ match area and call on Matchtri()
Matchtri( )	➔ compute the 3 sides and call compute3()
Compute3( )	➔ check if SSS theorem is met
Compute2( )	➔ check one side of the triangle

Graphical User Interface:

In order to display the world coordinates of the Cycab, a GUI is created. In the first window (refer to appendix A) known as "world SLAM", the absolute coordinate of the Cycab's position will be displayed before world localization is completed. Once world localization is completed, the GUI notifies the user of the detection of known fixed landmarks and allows the user to choose to display either the absolute coordinate or the world coordinate of the Cycab's position in the car park. This position coordinate displayed will be updated concurrently with the motion of the Cycab. A second GUI window known as "worldSlamServ.elf" is constructed to display the verified landmarks in the currentMap. This will provide the user with visible view of the adding of the verified landmarks. The main function is for the programmer to check for the performance of the world localization. Information that the programmer can obtain are given in the following example.

Refer to appendix A, there are two verified landmarks highlighted by a red circle each. These two landmarks cannot be found under "Top view" hence, the programmer can know that these two landmarks are actually error landmarks due to the noise during detection by the SICK. This error that is due to noise cannot be fully eliminated but it can minimize by placing newly discovered landmarks into waitingMap until it is seen for a define number of time before it can be verified and moved to the currentMap. The black area highlighted by another red circle should have two landmarks. They are not displayed because the Cycab's sensor had yet to discover them. If all or most of the landmarks' location in the "worldSlamServ.elf" are different from those in "Top View", the programmer will know that the world localization has failed. (Refer to appendix A)

Performance evaluation:

Result of the experiment shown that the Cycab is able to match the correct triangle and compute its location in the world frame when the following conditions are met:

1) The Cycab sees the fixed landmarks
2) The position of the fixed landmarks are unique
3) SLAM is being done correctly

Current work:

The basic function of the world localization thread is done. However, the computation of the transformation variables from absolute to world frame is fully dependent on the accuracy of the estimated Cycab's position from SLAM thread. Hence, if the SLAM fails to produce correct result, the world mapping will be incorrect. Cases that cause SLAM to failed are:

1) SLAM produced a wrong estimation of Cycab's position when the Cycab is in an area without any landmark where SLAM fully depends on the odometry data to compute its position.
2) The Cycab drifted from it actual motion path.

Currently the codes of the SLAM are being looked into to find the problem resulting to the situation above.

Other possible improvement of the program is also being looked into, they are:

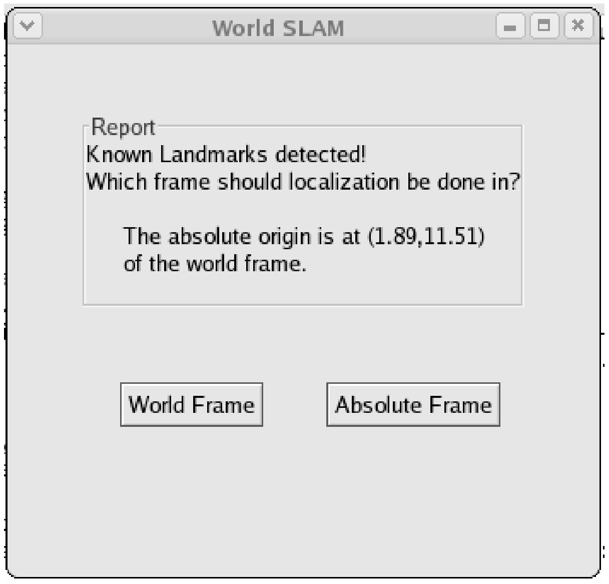
1) Creating a database that contains triangles properties of landmarks that are found in the currentMap. Using this database for matching instead of the AMap.
2) Instead of stopping world localization once a match is found, it is aim to perform world localization's matching each time the currentMap is being updated.

References:

1) Cedric Pradalier and Sepanta Sekhavant. Simultaneous Localization and Mapping using Geometries Projection Filter and Correspondence Graph Matching

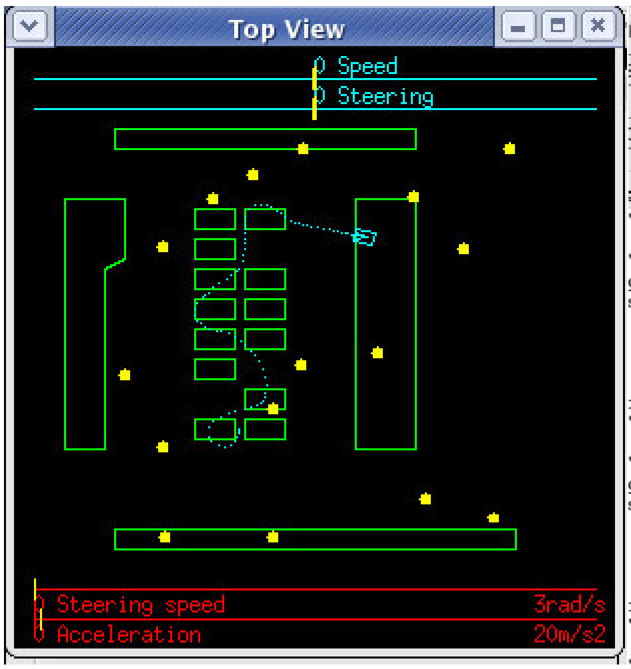
2) Eric W.Weisstein. http://mathworld.wolfram.com/Triangle.html

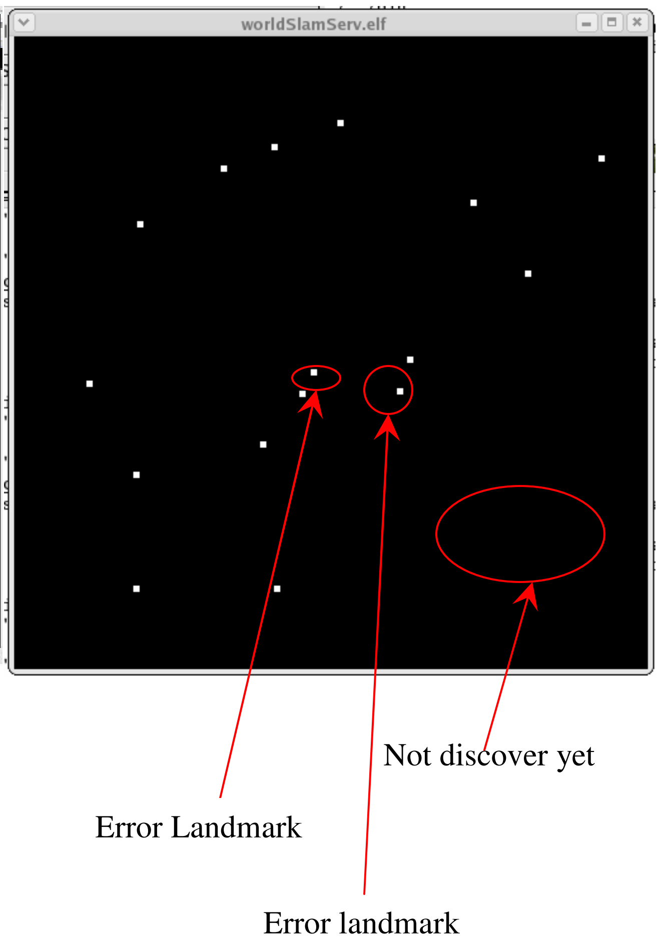
Appendix A



Report
Known Landmarks detected!
Which frame should localization be done in?

The absolute origin is at (1.89,11.51)
of the world frame.


Top View
0 Speed
0 Steering
0 Steering speed
3rad/s
0 Acceleration
20m/s2


worldSlamServ.elf
Not discover yet
Error Landmark
Error landmark