

Available online at <http://www.mecspress.net/ijwmt>

Comparative Study of Navigation Methods for Unmanned Vehicles in a GPS-Denied Environment

Sushmita S Warriar^a, Hardik Modi^b

^aUndergraduate Student, Charotar University of Science and Technology, Changa-388421, Gujarat, India

^bAssistant Professor, Charotar University of Science and Technology, Changa-388421, Gujarat, India

Abstract

In this era of increasing technological advancement, where mankind is pushing the frontiers of exploration, systems are required to be increasingly autonomous and intelligent in order to work efficiently, and to provide optimum results. Unmanned vehicles generally use GPS to find their position in an environment and navigate. In this paper, we have studied and analysed two different algorithms used for navigating in a GPS denied environment, which is a more common scenario than navigating in a GPS-enabled environment. The SLAM algorithm generates a map of its environment by estimation and mapping, while the LIDAR maps its environment using pulses of a short wavelength. Both of them can be used for widely varying applications, and current researches in these fields are crucial in the development of autonomous systems.

Index Terms: GPS-denied environment, SLAM, LIDAR, navigation, unmanned vehicles, submap, orthophoto.

© 2016 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

1. Introduction

Unmanned missions are meant for reconnaissance, survey and search and rescue missions in places too remote or dangerous for humans to physically perform these tasks. Some examples of such locations can be a calamity-hit area, inhospitable mountainous regions, or on a battlefield. The most prevalent navigation method used for unmanned tasks today is GPS (Global Positioning System). GPS requires an unobstructed line of sight from the position of the object/robot to four or more GPS satellites. Hence, GPS is available and usable only when there is a clear sky. It fails to operate where it is impossible or difficult to receive satellite signals, and even when it is present, its accuracy is insufficient for precise unmanned navigation and mapping- the error margin of GPS/GNSS (Global Navigation Satellite System) is 10 meters on average. This has rendered these systems unusable in cases where precision is of utmost importance.

* Corresponding author.

E-mail address:

To solve this, research has focussed on alternate techniques, the most notable being Simultaneous Localization and Mapping (SLAM), Light Detection and Ranging (LIDAR), and Radio Frequency Identification (RFID). Here we review the navigation systems based on LIDAR and SLAM. Both these algorithms are fundamentally different in their implementation and usage, and hence this paper aims to provide a comprehensive understanding on the contrast in their execution. First, we have explained the reason for these algorithms to be implemented. Then, we have introduced the basic concepts of the algorithms, and given a description of its working, different aspects of the algorithms and methods used to improve their accuracy and efficiency.

2. Slam

Simultaneous localization and mapping (SLAM) is a navigation technique that uses mapping and estimation to build a map of the robot's environment. Theoretically, SLAM can be implemented in ground, aerial, underwater and indoor robots. Practically however, it is still slightly difficult to generate a detailed extensive map, because of the huge processing overhead involved. Again, there are various forms of SLAM available-the extended Kalman filter (EKF) SLAM, 3-Dimensional SLAM and the trajectory-oriented SLAM are just few of the examples of the research results in this field.

2.1. Introduction to SLAM:

The SLAM algorithm tries to determine if it is possible to place a robot in an unknown environment at an unknown location and to have it determine its location by building a consistent map of its surroundings. Making SLAM highly efficient and accurate will provide a means to make a robot completely autonomous. It involves estimating the location of landmarks and trajectory of platform online without the need of any prior knowledge of the location.

In a 1986 IEEE Robotics and Automation Conference, it was recognized that a fundamental problem in robotics was consistent probabilistic mapping. Papers published by Durrant-Whyte [1] and Smith and Cheesman[2] established a statistical basis to describe relations between manipulating geometric uncertainty and landmarks[3]. It showed that there must be a high degree of correlation between the estimates made by a robot on the locations of different landmarks in a map. It also showed that these correlations would only grow with each successive observation [3].

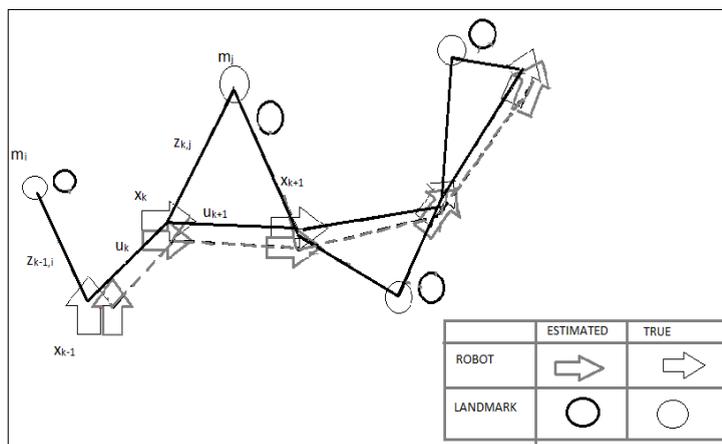


Fig.1. The SLAM problem Observations are made between true landmark and robot locations, instead of being measured directly.[3]

An important paper by Smith [4] indicated that as a robot travels in an alien environment taking relative observations of the landmarks in its surrounding, their estimates are all necessarily correlated due to the common error in estimated robot location [5]. This meant that the joint state comprising of vehicle pose and landmark position would have to be updated following each observation; leading to employment of a huge state vector with computational scaling as the square of number of landmarks.

In the earlier analysis of these problems, the convergence properties or steady state behaviour of the map were not looked at. Due to this, the correlations were forced to be minimized or eliminated, reducing the filter to a series of decoupled landmark to vehicle filters [6],[7]. The breakthrough was the realisation that the combined and localization mapping problem were actually convergent, and the efficacy of the solution increased with the number of correlations. In fig. (1):

\mathbf{x}_k is the state vector describing orientation and location of robot.

\mathbf{U}_{k-1} is the control vector applied to drive robot from \mathbf{x}_{k-1} to \mathbf{x}_k at time k .

\mathbf{m}_i is the location description vector of a time-invariant(assumed) i th landmark.

z_{ik}/Z_k is the observation taken from the robot at time k of the location of the i th landmark. When there are multiple landmark observations at the same time, it is denoted as Z_k .

$X_{0:k}$ is the set of history of robot locations = $\{x_0, x_1, \dots, x_k\}$

$U_{0:k}$ is the set of history of control inputs = $\{u_1, u_2, \dots, u_k\}$

M is the set of landmarks = $\{m_1, m_2, \dots, m_n\}$

$Z_{0:k}$ is the set of landmark observations = $\{z_1, z_2, \dots, z_k\}$

In probabilistic SLAM, it is required that the probability distribution $P(\mathbf{x}_k, \mathbf{m} / Z_{0:k}, U_{0:k}, \mathbf{x}_0)$; which describes the joint posterior density of the landmark locations and the robot state, when the recorded observations and control inputs up to time k are given along with the initial state of the vehicle; must be computed for all times k . Here the map building problem may be formulated as calculating the conditional density $P(\mathbf{m} / X_{0:k}, Z_{0:k}, U_{0:k})$, which assumes that the robot location \mathbf{x}_k is known always, subject to knowledge of initial location. In contrast, the localization problem assumes that landmark positions are known with certainty and vehicle positions are to be determined [3].

It is important to note that the landmark observations correlations increase monotonically with every successive observation. Due to this high correlation, when a robot observes a landmark \mathbf{m}_j from a location \mathbf{x}_k , it updates all previous landmarks along with the current landmark. This leads to a more accurate map building, even if the absolute position of the robot with respect to the landmarks is not known.

2.2. State augmentation and partitioned updates:

State augmentation is a technique to limit time-update computation, while partitioned update is a technique to limit observation-update computing.

A joint state vector consists of robot pose and \mathbf{m} , which is the set of landmarks. The vehicle model propagates the pose states according to a set of control inputs while leaving the map states unchanged [8]. The operation which is used to compute the covariance prediction has cubic complexity in the number of landmarks. However, it can be rewritten in a form that has linear complexity, taking advantage of the fact that only pose states are affected by the vehicle model [8], [9]. To add a new landmark to the vector, it is initialized as a function of the observation z_k and the robot pose. These augmented states then become a function of a few existing states, namely: \mathbf{x}_{vk} , which is the robot pose at time k , \mathbf{m} , and $\mathbf{g}(\mathbf{x}_{vk}, z_k)$, which is the function by which a new landmark is initialized.

A partitioned update method reduces the complexity of computation unlike the EKF (Extended Kalman Filter) update, where because the vehicle and map states are updated after every measurement, the computational complexity increases quadratically with respect to the number of landmarks. It achieves this by

updating the global map at a much lower frequency and confining sensor-rate updates to a small local region [8].

CEKF (Compressed EKF) [10] and postponement algorithm [11] operate only in a local region, while keeping globally referenced coordinates. On the other hand, the CLSF (Constrained Local Submap Filter) [9] and local map sequencing algorithm [12] generate a short-term submap with its own local coordinate frame [8], a simpler approach which avoids large global covariances by performing operations in the local coordinate frame at a high frequency, thus maintaining stability and being less sensitive to linearization errors. The updates in a conventional SLAM are performed only on the landmarks held in the local submap. A favourable global estimate can be found by registering the submap with the global map, and a global pose estimate may be obtained by a vector summation of the local referenced pose and a global estimate of the submap coordinate frame [8].

The submap methods are of two types: globally referenced and locally referenced. In both of them, a local coordinate frame is defined by the submap and landmarks are estimated with respect to that frame. Global submap methods estimate the global locations of submap coordinate frames relative to a common base frame [8]. The relative landmark representation [13], hierarchical SLAM [14] (named after the local submap structures being arranged in a hierarchy) and constant time SLAM (CTS) [15] all adopt this approach.

However, the relative submap offers more advantages-computational complexity is independent of the size of the map, and because it treats updates locally, it allows features like batch association, and reduced linearization errors without compromising on the optimality.

2.3. Data association and batch validation:

During the implementation, any new landmarks are first associated with existing landmarks, and then the data is fused into the map. After fusion, this data cannot be revised. Hence, even one incorrect association can cause a divergence in the map, leading to a failure of SLAM. The batch gating technique is used to avoid this by considering multiple associations simultaneously. The combined constraint data association [16], a graph search, can do so reliably without any knowledge of vehicle pose.

However, for more complex environments, more comprehensive techniques (such as multihypothesis tracking [17]) are required. Multihypothesis tracking generates separate track estimates for each association, creating over time a branching tree of tracks [8]. A huge computational overhead is incurred in maintaining separate estimates, but submap or sparsification methods may be used to provide an amenable solution.

3. LIDAR

LIDAR is used to efficiently create terrain models digitally and with high accuracy. This technique is especially useful to map large regions, while allowing for very detailed 3D mapping. This includes the terrain of an area, as well as objects on the terrain surface. LIDAR is thus very suited for use in mapping dense areas such as forested areas or urban regions.

3.1. Introduction to LIDAR:

LIDAR (Light Detection and Ranging) works by mapping the surface by measuring reflected laser pulses using for e.g., an oscillating mirror. Similar to RADAR, it however emits short wavelengths in place of radio waves. An important feature of LIDAR is that it can differentiate between first and last pulse reflection. This is extremely beneficial because the first reflection is generally off the tops of building or trees, while the last reflection is more likely to represent the ground terrain.

Additionally, derivation from reflected radiation provides intensity images, which are recorded in different shades of grey (for buildings and trees) while the water surfaces are represented by white (because there is no laser reflection to be recorded).

Accuracy of LIDAR is 20 to 30cms in vertical and horizontal plane. The accuracy of LIDAR in horizontal plane is subject to ground slope and altitude conditions.

3.2. What is true orthophoto?

An orthophoto, or any aerial photo for that matter, is affected from radial displacement. This means that elevated objects like buildings and hills will be displaced radially outwards from the centre [18,19]. An orthophoto factors in Digital Terrain Modeling (DTM) during the rectification process to remove this radial displacement, which happens due to perspective projection. Perspective projection is the image that mimics how humans view 3D objects in a linear plane. Because of the rectification, the orthophoto is projected in a parallel fashion, at a constant scale. But because objects like buildings are not modelled in DTM, they are distorted, in the form of leaning buildings and warped bridges [19,20].

True orthophoto solves this quandary by factoring DSM (Digital Surface Modeling) instead of DTM during rectification.

3.3. Rectification:

As mentioned earlier, relief displacement causes objects at different heights are projected to different positions in the photograph, i.e., objects at a higher position appear bigger in the photo. The ortho rectification process aims to remove this perspectiveness of the image. It ensures an orthographic projection with parallel rays, instead of being perspective, where rays pass through a common perspective centre. This is demonstrated in fig 2 [22, 23].

By doing this, we can obtain an orthophoto with constant scale, which can then be used for combining with spatial data in GIS systems, or as part of 3D visualisations where the orthophoto is draped over a 3D model [22]. The fact that the orthophoto has a world coordinate reference means that it may be used as an uninterpreted map.

3.4. Mosaicking:

Generally, orthophotos of overlapping areas are merged into one orthophoto of a large area. This is called mosaicking. It involves three steps:

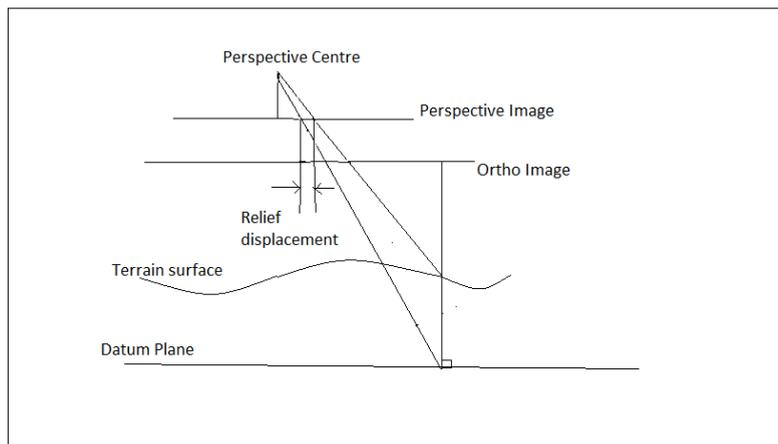


Fig.2. Orthographic and perspective images, indicating the cause for relief displacement [19].

1. Seam line generation
2. Colour matching
3. Feathering and dodging [22]

Seamlines are the contours in an orthophoto which indicates the merging of photos. It can be automatic or manual. Seam lines are placed in portions where the overlapping images look very similar. Manually, it is generally placed along the centre of roads. Automatically, one method is to place seamlines along the centre of the overlapping areas. Another approach is when the photos are subtracted and the seamlines are placed along the difference between the two images.

An important point to be noted while creating the mosaic is that since images with different radiometry and colour characteristics are considered, the seamlines need to ensure that orthophotos are fused such that the mosaic does not look like it has content from several images, failing which the mosaic could appear patchy.

Colour matching analyses and compares overlapping regions to remove the radiometric differences. Dodging techniques perform the same function. Feathering performs the task of smoothing the overlap so that the cut smoothly merges from one image to another.

There are basically two methods to generate true orthophotos: automatic, for when the surface is represented properly; and semi-automatic, for when objects cannot be properly represented. In this paper we discuss both of these approaches.

3.5. Automatic approach:

Here, the DSM along with aerial images is factored in during the rectification. This is performed in two steps.

First, perspective projection produces image coordinates by interpolating the height values using DSM. Then grey levels from the input image are interpolated and assigned to the output image [19]. This is called image resampling, and the method used for this is bilinear interpolation. This method uses the distance weighted average of the nearest four pixel values to estimate a new value. Then a visibility map is created through a visibility check. If this is not done, there arises a problem called ghost imaging. This occurs when buildings are projected in occluded areas (blind spots) as well as on the correct place obtained from the DSM [19]. Finally, overlapping photos are merged to fill blind spots by running a mosaic process. Image information from adjacent orthophotos is used to fill occluded areas, and a true orthophoto mosaic is created based on the overlapping image seamlines. However, if not enough orthophotos are available for overlapping, all occluded areas may not be filled.

3.6. Semi-automatic approach:

In the above approach, details such as ridges and roof borders are not properly represented. First of all, instead of a surface texture, a terrain texture is extracted. This avoids the building and trees representation. Then, footprints of the removed objects are filled with a regular grid based on the specified density parameter during production of a DTM.

The bilinear interpolation method is used here, just like in the automatic approach. Then, roof borders and ridges are digitized using a photogrammetric work station and superimposed on the DTM to make DSM [19]. The final step is to create a mosaic and a true orthophoto as mentioned in the previous approach.

3.7. Surface representation:

The surface of a terrain is most commonly represented by contours in a map. Each contour line represents a constant height of the terrain. The hillier the terrain, the closer the lines are to each other. There are generally two ways to represent surfaces for data processing. One is the TIN method and the other is the grid method.

TIN (Triangulated Irregular Network): It comprises measuring a series of heights through the surface, to which a network of triangles is connected. A rough surface description is obtained by interpolating the points between the vertices of the triangle enclosing the particular point in the network.

To connect the points in a network of triangles, we look at the Delaunay triangulation algorithm. This method tries to connect the points so that it maximizes the minimum angles in all triangles, along with having the smallest possible total edge length [24, 22].

A limitation of this method is that it is unable to handle vertical or overhanging objects, since, at the same point, it requires more than one value of height. It provides a more comprehensive description of buildings, which the grid method cannot achieve without a highly dense grid.

Grid: This method regularly spaces points in the x and y directions. It is comparable to a matrix, wherein the height in the cell is represented by the cell. Since it is extremely difficult to measure all heights in the grid, it is generally built on the basis of datasets, for example TIN or contours.

This grid method has the advantage of computational ease, because of the grid being handled as a matrix. Also, it is easier to determine height at a given point, as only the nearest grid points need be located and the required point interpolated. However, accuracy is limited due to the detailing being limited to grid size. This is illustrated by the fact that the grid size does not allow rapid changes in elevation.

4. Conclusions

With the increase in the usage and development of unmanned vehicles, it becomes necessary to develop an approach to map the surroundings as accurately and quickly as possible, without compromising on the quality or increasing the cost of implementing such a procedure.

While the SLAM method provides a way to map the robot's environment by measuring its distance and pose relative to the landmarks in its surroundings, the LIDAR approach proposes a way to map the environment by capturing aerial images and generating true orthophotos, modelled on DSM. Algorithms for both the SLAM and LIDAR techniques have been discussed. This means that the SLAM algorithm is better suited for application in areas where an imaging of the environment is not required, for example in fully autonomous reconnaissance missions, while LIDAR is better suited for applications where a precise imagery of the surroundings is required, for example in survey missions. Both of these techniques have high potential and with the amount of ongoing research in these fields, a cheap and perfectly accurate map of the environment will soon be possible.

Acknowledgment

We would like to thank Charotar University of Science and Technology for support provided.

References

- [1] H.F. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE Trans. Robot. Automat.*, vol. 4, no. 1, pp. 23–31, 1988.
- [2] R. Smith and P. Cheesman, "On the representation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1987.
- [3] Durrant-Whyte, Hugh, and Tim Bailey. "Simultaneous localization and mapping: part I." *Robotics & Automation Magazine, IEEE* 13.2 (2006): 99-110.
- [4] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I.J. Cox and G.T. Wilfon, Eds. New York: Springer-Verlag, pp. 167–193, 1990.

- [5] J.J. Leonard and H.F. Durrant-Whyte, "Simultaneous map building and localisation for an autonomous mobile robot," in *Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS)*, Osaka, Japan, 1991, pp. 1442–1447.
- [6] J.J. Leonard and H.F. Durrant-Whyte, *Directed Sonar Navigation*. Norwell, MA: Kluwer, 1992.
- [7] W.D. Renken, "Concurrent localization and map building for mobile robots using ultrasonic sensors," in *Proc. IEEE Int. Workshop Intell. Robots Syst. (IROS)*, 1993.
- [8] Bailey, Tim, and Hugh Durrant-Whyte. "Simultaneous localization and mapping (SLAM): Part II." *IEEE Robotics & Automation Magazine* 13.3 (2006): 108-117.
- [9] S.B. Williams, "Efficient solutions to autonomous mapping and navigation problems," Ph.D. dissertation, Univ. Sydney, Australian Ctr. Field Robotics, 2001.
- [10] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implementation," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 242–257, 2001.
- [11] J. Knight, A. Davison, and I. Reid, "Towards constant time SLAM using postponement," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 2001, pp. 405–413.
- [12] J.D. Tard'os, J. Neira, P.M. Newman, and J.J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 311–330, 2002.
- [13] J. Guivant and E. Nebot, "Improving computational and memory requirements of simultaneous localization and map building algorithms," in *Proc. IEEE Int. Conf. Robotics Automation*, 2002, pp. 2731–2736.
- [14] C. Estrada, J. Neira, and J.D. Tard'os, "Hierarchical SLAM: Realtime accurate mapping of large environments," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 588–596, 2005.
- [15] J. Leonard and P. Newman, "Consistent, convergent, and constant time SLAM," in *Proc. Int. Joint Conf. Artificial Intelligence*, 2003.
- [16] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," Ph.D. dissertation, Univ. Sydney, Australian Ctr. Field Robotics, 2002.
- [17] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.
- [18] Thorpe, A., (2001) *Digital orthophotography in New York City*, [Online], http://www.sanborn.com/Pdfs/Article_DOI_Thorpe.pdf
- [19] Günay, Arif, Hossein Arefi, and Michael Hahn. "True Orthophoto production using Lidar data." *Joint Workshop "Visualization and Exploration of Geospatial Data", International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. 36. 2007.
- [20] Schickler, W. and Thorpe, A., *Operational Procedure for automatic true orthophoto generation*, ISPRS Commission IV Symposium on GIS - Between Vision and Applications, Vol. 32/4, Stuttgart, Germany, 1998
- [21] BRAUN, JOSEF. "INPHO GMBH, Stuttgart." *Aspects on True-Orthophoto Production* (2003).
- [22] Nielsen, Morten Ødegaard. *True orthophoto generation*. Diss. Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2004.
- [23] Mikhail, E.M, et al, 2001, *Introduction to Modern Photogrammetry*, Wiley and Sons Inc.
- [24] Overmars, M., et al, 2000: *Computational Geometry*, 2nd edition, Springer-Verlag.

Authors Profiles



Sushmita Warriar is currently pursuing B.Tech from Charotar University of Science and Technology, Changa, Gujarat. Her research interests lie in intelligent robotics, embedded system design, intelligent automation and man-machine interaction.



Hardik Modi has received his B.E in Electronics and Communication Engineering from Sardar Patel University, and M.E degree in Electronics and Communication System Engineering from Dharamsinh Desai University. Presently, he is Research Scholar at Charotar University of Science and Technology. He is also working as Assistant Professor in Electronics and Communication department at Charotar University of Science and Technology, Changa, Gujarat. His research interests include microcontroller applications, medical imaging, embedded system, image and video processing and computer vision.

How to cite this paper: Sushmita S Warriar, Hardik Modi, "Comparative Study of Navigation Methods for Unmanned Vehicles in a GPS-Denied Environment", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.6, No.2, pp.30-38, 2016.DOI: 10.5815/ijwmt.2016.02.04