Inertial Navigation Algorithms

Ken Mulder, Maiga Chang, Larbi Esmahi
School of Computing and Information Systems
Athabasca University
Athabasca, Canada
kenwmulder@hotmail.com,
maigac@athabascau.ca, larbie@athabascau.ca

Mohamed Jemni
Research Laboratory of Technologies of Information and Communication & Electrical Engineering
University of Tunis
Tunis, Tunisia
mohamed.jemni@fst.rnu.tn

Abstract—The main goal of this research is to design an Inertial Navigation System (INS) which can effectively position a user by using a monocular camera with fiducial markers. People have become accustom to the ease of navigating with Global Navigation Satellite Systems (GNSS) but the system fails indoors when signals degrade. This paper describes the work in progress of the system design, the algorithms in particular, to provide reasonably accurate, precise and reliable indoor navigation.

Keywords—inertial navigation; fiducial marker; spatial cognition; augmented reality; agent-oriented programming.

I. INTRODUCTION

"Are we there yet?" is a trivial question most kids ask too often to the annoyance of their parents, but it is a nontrivial question depending on the context of who is asking the question and what the situation is when the question is asked. For instance, the question is important if it is a handicapped person trying to find the closest emergency exit in a fire especially if the fire and smoke blocks the most immediate exit. As well, it may be a nontrivial solution to determine a precise interior location and direction to the destination because Global Navigation Satellite Systems (GNSS), cellular and wireless signals are degraded, jammed or unavailable inside of buildings.

A combination of devices need to be utilized to determine position inside tight interior spaces [11]. Wu et al. designed a mobile learning application using a landmark or allocentric navigation system (ANAV) for mobile devices that have neither a built-in GNSS receiver nor a compass [17]. They suggested the following four primary issues to enhance the functionality and increase the performance of the application: shortest path, object shaping, obstacle avoidance and visibility. A fifth issue is related to reducing the cumulative inertial navigation system (INS) error by correcting the three-dimensional (3D) position using a monocular camera and fiducial markers.

An application could identify a user's positions better when he or she takes a picture of the fiducial marker. The user's position could be found based on the transformation of the fiducial marker's shape and the result compared to the record stored in the cognitive spatial data model. This research will address these issues by enhancing the ANAV with fiducial markers, augmented reality, spatial-temporal data model and spatial learning tasks.

The research objective is to enable navigating indoors using a hand held Android device. Section II reviews relevant literature in spatial cognition to identify important problems in the field. The review will lead to the reasons for the basis of this research project. Section III analyzes the path generation, orientation and navigation system tasks. Section IV discusses the detailed algorithm design for the system tasks. Section V briefly talks about the design and development of the system architecture. At the end, section VI concludes with a discussion of future research directions.

II. RELEVANT RESEARCH

The Android based system developed by Wu et al. utilized Quick Response (QR) code markers to determine the user's position [17]. The user could replace the pre-stored and pre-calculated position with the more accurate position of the QR code. This research will also investigate using a camera to scan combined landmarks, fiducial and QR code markers to determine the 3D position. Fiducial markers are more effective at determining position because their design is simpler than QR codes and that increases the range and precision for calculating the relative positions of the marker and camera [15].

Our research uses Augmented Reality (AR) at the fiducial marker orientation points and not while moving because it has been shown to have mixed success in transit when sub-meter positioning is difficult to achieve [4]. Augmented reality can be used to make map interpretation less difficult by removing the mental map rotation and map symbol correlation to the surrounding real world.

An egocentric navigation system (ENAV) [5] can be added to the ANAV. ANAV is based on external direction references such as the cardinal directions north, south, east and west while ENAV is based on internal direction references such as the relative directions left, right, forward and backward. The ENAV will use an INS to navigate with dead reckoning or path integration to determine the user position as they move. Path integration is the integration of the motion senses or sensors along paths to determine the direction and distance from a starting point. Darwin [3] noted the ability for animals to dead reckon and Murphy [13] referred to this as an integrating process. This would be referred to later as path integration [12] to denote the integration of motion senses to sense the path direction back to the original starting point.

978-1-5090-4338-5/17/$31.00 ©2017 IEEE

III. Analysis

The first system task is path generation. The Navigation Mesh (NavMesh) is generated with the Project Tango tablet scan [6] by building voxels, 3D pixels, from the 3D scanned point cloud. This is done in three steps: rasterization into voxels, extraction of walkable surfaces, and generation of a NavMesh. The A* algorithm can be used to determine all the possible paths between destinations across the NavMesh by selecting the closest visible corners of a corridor (Fig. 1).

The proposed path generation improves on the performance of the basic directional algorithm used by Wu et al. [17] to calculate student paths to the learning activity destinations. The algorithm determines navigation paths well within the tight confines of indoor floors, hallways and doors. This algorithm efficiently navigates the spatial-temporal model. For instance, the algorithm determines where and when to go to minimize time navigating and maximize time learning. The utility of learning at particular times is increased or decreased by also recommending times to go or not go to a destination such as during docent presentations or not during busy times respectively.

This is controlled using navigation areas, costs and masks which can be tuned to simulate busy and obstructed areas. Costs can be applied in advance to areas to make agents take longer to pass through to model situations where the fastest path may not be the shortest path. A mask can even be applied to an area during runtime to signified an obstruction, such as a fire, to prevent a regenerated path from proceeding through that area.

The second system task is orientation. The position and pose determined while scanning an AR marker at a control point will be used to initialize the orientation algorithm. The first step for all users will be a path integration test relative to the last landmark or the start marker when initializing the first time. This involves pointing the camera in the direction of the marker and entering the number of steps of the direct distance to the marker regardless of any obstacles.

In the second step the system will show the user the path and its description. The path uses standard International Orienteering Federation (IOF) map symbology [8] with red triangles, circles, and double circles representing the start, control and finish markers respectively connected by a red path. The cognitive spatial data model presented in the map will display a simplified view of the boundaries and landmarks to reduce unnecessary details or distractions. For instance, it avoids the need to display an obstruction by just displaying the navigable path around the object. Detail images will be provided for important landmarks such as the orange and white IOF control and fiducial markers. Images of existing signs can also be added to provide important directional information.

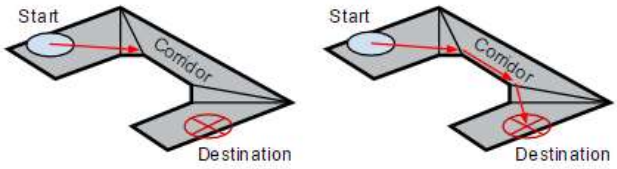

Figure 1. NavMesh corridor path [16].

The third system task is navigation. The navigation interface will display navigational information based on the orientation. The view includes an IOF control marker description with textual description for users who are unfamiliar with the IOF symbology. This interface will then display the user's position based on the sensor readings processed through a complementary filter algorithm. While the extended kalman filter is the defacto standard method for sensor fusion, the complementary filter requires less processing resources to produce similar results which makes it more suitable for mobile applications [10]. Fused gyroscope and accelerometer sensor readings will be recorded as the INS positions. These positions along with the GNSS and AR positions will be recorded to determine the percentage of the corrected INS positions that are within the desired range of +/-3m. The performance of the proposed system will also be compared to the accuracy and precision of other indoor positioning systems [7].

IV. Algorithm Design

This section discusses the algorithm design for path generation, orientation and navigation. For example, in the path generation if there are 10 destinations (Line 1) in the environment then there will be 180 paths between all the destinations, 90 for walking and 90 for rolling if the user is in a wheelchair (see Algorithm 1). There will be 1 software agent starting at each of the 10 destinations and each agent will be assigned the destination of one of the 9 other destinations (Lines 2 and 3). The agents will first generate a walking path and then generate a rolling path for wheelchairs to avoid stairs (Line 4). The path generation maybe different depending on the direction between destinations because some paths are unidirectional (e.g. turnstiles, escalators, etc.). The distances and sprites for each path will be saved to be used later for spatial cognition (Lines 6 and 7).

ALGORITHM 1: Path Generation

Input: Agent[walking, rolling], Destination[]
Output: Distance[] and sprites
1: **For** Destination[i] = 1 **to** 10 **do**
2: **For** Destination[j] = 1 **to** 10 **do**
3: **If** j<>i **do**
4: **Foreach** Agent[k]
5: Create path from Destination[i] to [j] for Agent[k]
6: Calculate Distance[i,j,k]
7: Create sprite and add to atlas
8: **EndForeach**
9: **EndIf**
10: **EndFor**
11: **EndFor**
12: **return** Distance[] and sprite atlas

Algorithm 2 shows the process for determining the orientation with the spatial cognition level and the appropriate map view. The path and its landmark description are created as the path buffer is generated around landmarks on the NavMesh. The path is generated by an A* algorithm. The path starts with the current user position and moves to the selected destination position (Lines 1 to 7). The landmark description includes

symbols and text in relative and cardinal directions. For example, if a user is heading west towards a landmark in the northwest, then the cardinal direction symbol will point to the upper left but the relative direction symbol will point to the upper right. Lines 8 to 10 print out the completed landmark description after the path is generated. Lines 11 to 15 display, run and score the spatial cognition learning tasks which are used to evaluate the user level. The scores are marked in a range from 0 to 4 where 0 is the least accurate and 4 is the most accurate.

ALGORITHM 2: Orientation

Input: landmarks, user position, task sprites, NavMesh
Output: spatial cognition level
1: Set the buffer zone
2: Select a destination landmark from landmarks
3: **While** generating a top view map from the user position
4: **If** landmark is within buffer
5: add to landmark description[]
6: **EndIf**
7: **EndWhile**
8: **Foreach** landmark description[]
9: Print landmark description[]
10: **EndForeach**
11: **Foreach** spatial cognition learning task[]
12: Display task[]
13: Run task[]
14: Score task[]
15: **EndForeach**
16: Spatial cognition level = average task score
17: Generate a map view appropriate for spatial cognition level
18: **return** spatial cognition level

For example, in the path integration test a user's sense of direction and distance is evaluated based on their perception of the last and next destinations. The user can choose meters or their own step length as a unit of measure. For instance, if the user selects steps, then the distance (e.g. 10 steps) is converted to meters before scoring the user's distance estimate. The distance is scored from 0 to 4 and the score is inversely proportional to the difference between the user's estimate and the actual distance. If the user estimates 5 meters and the distance is 5 meters, then he or she would score 4 but if the estimation is 10 meters or more, then he or she would score 0. An estimation of 2.5 meters or 7.5 meters on the other hand would score 2. Similarly, the direction is also scored depending on how close the user's estimate by pointing the camera in the direction of the destination is relative to the actual direction. If the user estimates north and the direction is north, then they would score 4 but if the user estimates south then they would score 0. In the same case, an estimate of east or west would score 2.

The uniformed field of view and mental rotation tasks test a user's sense of direction and ability to recognize the path respectively. Both tasks are scored in a similar fashion to the path integration test on a scale from 0 to 4. Finally, the scores are averaged to determine the user's spatial cognition level (Line 16) and the level is used to determine the map view (Table 1) appropriate for the user (Line 17). The user should be motivated

TABLE 1 SPATIAL COGNITION LEVEL VIEW PROPERTIES

	View	Orient	Overview	Text	Route
0	Top	Forward	Distance/Direction	Relative	Curved
1			Direction		
2		North	Compass	Cardinal	
3				None	
4					Straight

to score higher and then be challenged by a map view which minimizes distraction by providing only the basic information the user requires. For example, a user scoring a spatial cognition level of 4 would be offered a north up view with a straight path which would require them to rotate the map to their environment and find the appropriate curved path around obstacles themselves.

The navigation algorithm starts with an initial pose from an AR marker (see Algorithm 3). The position and angle will be initialized from this pose (Lines 1 and 2). The velocity will be estimated based on the user's stride or push length per second (Line 3). The pose will be updated every 0.49 seconds unless the user pauses (Line 4). The pause is useful to prevent any drift in the calculations when the user is stationary. The time interval (Line 5) is the sampling rate of the gyroscope and it will be used to determine the change in the angle (Line 8) before the low frequency noise is removed by the high pass filter (Line 9). The accelerometer reading is adjusted, as well, to remove the acceleration from gravity before the high frequency noise is removed with the low pass filter (Line 10). The complementary results of the high and low pass filter (Line 11) are then added to create an updated attitude which in turn is multiplied by the estimated velocity to determine the change in position. The new position and attitude are then used to update the pose as represented by the frustum on the user interface (Lines 12 and 13). The drift will also be limited by constraining the pose to the navigable surface when rendering.

ALGORITHM 3: Navigation

Input: stride or push length, initialization pose
Output: pose
1: Position = position(initialization pose)
2: Angle = angle(initialization pose)
3: Estimated velocity = stride or push length/sec
4: Time constant T = 0.49sec
5: Time interval dt = 0.01sec
6: Filter coefficient a = T / (T + dt)
7: **Do while** navigation agent not paused
8: Angle = (angle + gyroscope * dt)
9: High pass filter = a * angle
10: Low pass filter = (1 - a) * accelerometer - gravity
11: Attitude comp filter = high pass filter + low pass filter
12: Position = position + attitude * estimated velocity
13: Pose = position + attitude
14: **EndDo**
15: **return** pose

V. The System Design and Development Plan

Multi-agent systems are ideally suited for designing and developing this system. The mental state of agent-oriented programming extends the biologically analogous concept of spatial cognition. This system architecture (Fig. 2) will connect distributed agents over the network using JAVA Agent Development framework (JADE).

JADE will be used for the distribution, communication, discovery and behavior of agents. Workflows and Agents Development Environment (WADE) extends JADE with a workflow engine and an Eclipse plugin IDE. WADE will be used for the development and administration of fault tolerant workflows. The WADE Workflow Status Manager Agent will persist workflow state in MySQL using Hibernate by deploying the WADE Persistence Add-On. WADE services will run on Apache Tomcat to provide a web service to dynamically allocate destinations to the users.

The Agent-based Multi-User Social Environment (AMUSE) variation of WADE will be implemented for user management including registration, authentication, communication and coordination. Communication between activities on an Android device will be passed with the intents. All development with JADE, WADE and AMUSE will be developed using Java Development Kit 1.7, Android Software Development Kit 5 and Android Native Development Kit r10e. The components configured in Unity will be scripted with C# and then built as an Android project to be imported into Android Studio to build the Android Application Package.

This research will select sections of the JADE methodology [14] based on the unified agent-oriented software engineering methodology [2]. This methodology should improve the quality of the design and reduce the time required to make it. To further improve the quality of the design and reduce time the research will utilize the Project Tango Development kit and ARToolKit to develop the mobile client. As well, the graphics will be abstracted to minimize the demands of collection and presentation. For example, AR requires less time and effort to build than virtual reality (VR) which is a significant impediment to implementing VR. This method has the additional benefit of presenting a less distracting interface to the user. These components of hardware and software will significantly decrease the time to produce high quality and quantity graphical input and output.

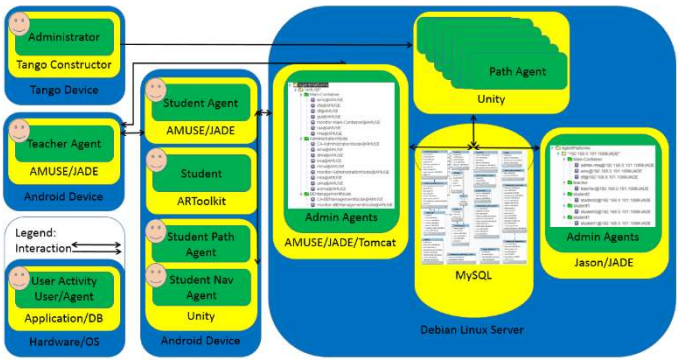

Figure 2. Multi-agent architecture.

VI. Conclusion

This low cost design could lead to a commercially viable solution that would be especially competitive during a period of economic recession when it is difficult to rationalize expensive hardware and software. Suggested areas for future research are to investigate implementing the system for use as an occupational therapy aid for people suffering injuries or diseases affecting their spatial awareness such as topographical disorientation disorder [9]. As well, if mobile 3D enabled devices become prevalent then a real-time spatial-temporal model combined with markerless AR will eliminate the need for model and marker preparation [1] and could provide an opportunity to develop the application for blind users. The Unity components could also be easily migrated to other platforms.

References

[1] Corbett-Davies, S., Green, R., & Clark, A. (2012). Physically interactive tabletop augmented reality using the Kinect. In the Proceedings of the 27th Conference on Image and Vision Computing, Dunedin, New Zealand, November 26-28, 2012, 210-215.

[2] Dama, H., & Winikoff, M. (2013). Towards a next-generation AOSE methodology. Science of Computer Programming, 78(2013), 684–694.

[3] Darwin, C. (1873). Origin of certain instincts. Nature, 7(179), 417-418.

[4] Duenser, A., Billinghurst, M., Wen, J., Lehtinen, V., & Nurminen, A. (2012). Exploring the use of handheld AR for outdoor navigation. Computers & Graphics, 36(8), 1084–1095.

[5] Fleming, P. (2005). Implementing A Robust 3-Dimensional Egocentric Navigation System. Vanderbilt University, Nashville, Tennessee, USA. unpublished.

[6] Google. (2016). Tango Constructor. Retrieved Aug 1, 2016, from https://developers.google.com/tango/tools/constructor.

[7] Hossain, M. & Soh, W. (2015). A survey of calibration-free indoor positioning systems. Computer Communications, 66(2015), 1-13.

[8] International Orienteering Federation. (2004). International Specifications for Control Descriptions. Retrieved July 29, 2016, from http://orienteering.org/wp-content/uploads/2010/12/IOF-Control-Descriptions-2004.pdf.

[9] Lim, T., Iaria, G., & Moon, S. (2010). Topographical Disorientation in Mild Cognitive Impairment: A Voxel-Based Morphometry Study. Journal of Clinical Neurology, 6(4), 204–211.

[10] Mahony, R., Hamel, T., & Pflimlin, J. (2008). Nonlinear Complementary Filters on the Special Orthogonal Group. IEEE TRANSACTIONS ON AUTOMATIC CONTROL, 53(5), 1203-1218.

[11] Mainetti, L., Patrono, L. & Sergi, I. (2014). A survey on indoor positioning systems. In the Proceedings of the 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, September 17-19, 2014, 111-120.

[12] Mittelstaedt ,M., & Mittelstaedt, H. (1980). Homing by path integration in a mammal. The Science of Nature, 67(11), 566-567.

[13] Murphy, J. (1873). Instinct: a mechanical analogy. Nature, 7(182), 483.

[14] Nikraz, M., Caire, G., & Bahri, P. (2006). A methodology for the development of multi-agent systems using the JADE platform. International Journal of Computer Systems Science and Engineering, 21(2), 99-116.

[15] Tsai, D. (2012). Autonomous Vision-Based Docking of the Tethered Axel Rover for Planetary Exploration. Luleå University of Technology, Luleå, Sweden. unpublished.

[16] Unity. (2016). Navigation Overview. Retrieved July 24, 2016, from https://unity3d.com/.

[17] Wu, M., Chang, M., Heh, J., & Kinshuk. (2011). Landmark-based Navigation System for Mobile Devices Have No Built-in GPS Receiver and Compass. In the Proceedings of the 10th World Conference on Mobile and Contextual Learning, (mLearn 2011), Beijing, China, October 18-21, 2011, 250-258.