Robust Autonomous Navigation in a Factory Environment

Chris Raabe[1], Niklas Bergström[2], Emad Saad[3], John Vian[3], Masatoshi Ishikawa[2] and Shinji Suzuki[1]

[1] Department of Aeronautics and Astronautics, The University of Tokyo, Tokyo, Japan

[2] Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan

[3] Boeing Research & Technology, the Boeing Company, Seattle, Washington, USA

***Abstract*:** *This paper describes the development and demonstration of a navigation system for a robot traversing a factory-like environment. The system is designed to combine several sensors to increase robustness and maximize the likelihood of successfully following navigation commands. The system uses motion capture, computer vision, and odometry sensors to form a robust estimate of robot motion. Performance degrades gracefully as some sensors become unreliable; however, the robot is able to continue safe operation despite multiple sensor failures. The system was deployed on a ground robot that was commanded to navigate through a course that was configured to represent a worst-case factory environment. The system was demonstrated at Boeing's Collaborative Autonomous Systems Lab in Seattle Washington.*

Keywords: *sensor fusion, computer vision, odometry, navigation*

1. Introduction

A prerequisite for robot navigation is the ability to identify the robot's motion in some known reference frame. There is a wide range of sensors that are employed to solve this problem. Some sensors are able to provide position and/or orientation information relative to a known reference frame directly. For example, the combination of GPS and a magnetometer can provide position and orientation measurements relative to the standard geodetic frame. Similarly, motion capture systems, such as Vicon [1], can provide both position and orientation measurements relative to a pre-decided reference that is fixed to the space in which the system is installed. Motion can then be determined by taking the derivative of those measurements.

Other sensors provide some position information, but relative to an unknown frame. Simultaneous localization and mapping (SLAM) is an often-employed technique for determining the location of the robot relative to some features in the environment. However, the location, orientation, and scale of those features may not be known a priori, in which case, only relative motion can be inferred.

Still other sensors can provide motion information, from which some position information can be inferred. For example, sensors like wheel encoders or visual flow can provide measurements of linear and angular velocity, which can be integrated to estimate the change in position and orientation from a previous time. However, this kind of position and orientation measurement is subject to integration drift. Furthermore, the position and orientation at a previous time might not be known.

All of these sensors are subject to some limits on reliability. GPS requires the ability to receive un-reflected satellite transmissions. Motion capture is prone to occlusion. SLAM is prone to ambiguity, map distortion, and mismatching. Odometry is prone to slip and drift. Our aim is to combine several sensors to increase overall reliability, particularly in the context of autonomous navigation in a factory environment. The difficulty in this approach is two-fold. First, identifying the quality of data reported from a particular sensor is not always straightforward. Second, when a sensor that reports relative position transitions from unreliable to reliable, the reference frame that it is measuring against may be unknown and must be aligned to a known reference frame. This paper discusses the approaches that we have taken to both problems, the set-up of a sample experiment, and the experimental results.

1.1. Outline

Section 2 discusses autonomous navigation in a factory environment, a few commonly-used sensors for such an application, and their shortcomings. In section 3 we propose a method for increasing the robustness of

http://dx.doi.org/10.17758/ER615302

autonomous navigation by combining several sensors, some of which may have unknown reference frames. Section 4 introduces an experimental setup that is representative of a worst-case factory environment. Results of that experiment are presented in section 5, followed by some concluding remarks.

2. Autonomous Navigation in a Factory Environment

As mentioned in the introduction, a prerequisite for robot navigation is the ability to identify the robot's motion in some known reference frame. There are many sensors that can satisfy this to some degree, but some are not suited to a factory environment. For example, GPS signals are not available and the dynamic environment makes outward-facing SLAM, for example with a scanning laser range finder, much more challenging. In this section, we discuss 3 sensors that we deemed to be suitable for a factory environment.

2.1. Motion Capture

Motion capture employs an array of cameras, each surrounded by a light source. The cameras detect light that is reflected back from passive retroreflective markers that are attached to the robot. The location of these markers can be triangulated when detected by 2 or more cameras, which have been calibrated to provide overlapping projections. When 4 or more markers are triangulated, the constellation of those markers can be matched to a vehicle whose constellation has been pre-registered with the system. Once matched, the position and orientation of that vehicle can be inferred from the position and orientation of the marker constellation.

This system is very popular for autonomous systems, such as small quadrotors (e.g. [2] and [3]), because it can provide very accurate position and rotation measurements at very high rates, while the computation is performed off-board. The downside however, is that reliable measurement requires enough markers to be visible to enough cameras for the constellation to be measured and matched. In the case of a factory environment, the robot might drive under a platform, or be subject to some other occlusion that would prevent accurate measurement.

2.2. Monocular SLAM

In monocular SLAM, features are extracted from an image stream from a single camera. These features are employed in 2 steps. First, the location of the camera is inferred by identifying features that match those in a database of previously discovered features and their locations (referred to as the map). Second, the inferred location of the camera is used to triangulate the location of newly discovered features and add them to the map. By this process, the system is able to provide drift-free measurements of the position and rotation of the camera, and by extension the robot that it is fixed to, without any knowledge of the environment. As the robot moves into an unexplored region, the system expands the map, but is able to recognize its location when it returns to a previously explored area.

Monocular SLAM works off of the assumption that the features it detects are mostly fixed to the inertial frame (i.e. not moving). This assumption might not be valid in a dynamic factory environment. To minimize this risk, we opted for a downward-facing camera, relying mostly on the features of the floor, with the assumption that moving objects will be mostly out of the camera's field of view.

Even without moving objects in view, monocular SLAM can be unreliable. The system might misidentify features and settle on an erroneous estimate of the camera location. This is particularly likely for self-similar textures, such as tile and brick. The system might not be able to detect enough features for reliable localization if the environment lacks texture, such as a white wall, or if the amount of light is too low. Finally, small error in triangulation or uncorrected distortion in the optics can lead to a distortion in that map that typically grows as the robot moves away from the origin.

2.3. Odometry

Odometry refers to the process of integrating motion information to form an estimate of position and/or orientation. For a ground robot, odometry can be performed using wheel encoders to infer linear motion from the rotation of the wheels. This requires knowledge of the diameter of the wheels and the assumption that they do not slip. Modelling error, integration error, and slippage generally lead to a decrease in accuracy of the estimate with distance.

3. Combining Sensors

The 3 sensors described in the previous section each have some conditions on the quality of the position and orientation measurements that they deliver. Ideally, these sensors should be combined in a way to maximize quality. However, there are two difficulties. First, identifying the quality of data reported from a particular sensor is not always straightforward. Second, when a sensor that reports relative position transitions from unreliable to reliable, the reference frame that it is measuring against may be unknown and must be aligned to a known reference frame.

3.1. Efficient Alignment

Of the 3 sensors discussed in the previous section, only motion capture gives measurements in a known reference frame. The reference frame for odometry depends on the robot's starting location and drifts as error accumulates. For monocular SLAM, the reference frame is assigned arbitrarily each time it is (re)initialized. Therefore, some method is required to align them. Furthermore, this alignment should only be performed when there is some confidence in the quality of the measurements that are being aligned against.

In a previous work, we proposed an algorithm to efficiently align a pair of sensors with differing reference frames [4]. The algorithm is based on Kalman filtering to estimate the coefficients that define the scale, rotational offset, and translational offset from one reference frame to the other. Emphasis is placed on characterizing the noise of measurements of each coefficient assuming that those measurements are derived from position and orientation measurements with Gaussian additive noise. The algorithm converges on the optimal transformation with speed that depends on variance of the noise in each of the measurements, and the motion of the vehicle.

The algorithm also produces a confidence measure for the transformation estimate, which can then be used to inform the fusion of the position estimates. That is, a sensor whose transformation to a known origin is not trusted should not contribute strongly to a fused estimate.

3.2. Quality Heuristics

Quality heuristics are required for 2 reasons. First, if a sensor is not able to provide reliable measurements, it should be abandoned in favor of a sensor with a higher quality measurement. Second, alignment of a sensor with an unknown reference to a known reference should only be performed when the sensor with a known reference is providing quality measurements.

Sensors are to some extent able to identify their own reliability, e.g. motion capture when it observes too few markers, or SLAM when the environment does not have enough features. However, even if they report that the measurements are good, they might be unreliable. Examples of such situations include the motion capture system incorrectly matching the registered markers to the detected markers resulting in a bad pose estimate, or the SLAM algorithm incorrectly matching the detected features to the map, resulting in a bad position or pose estimate. To cope with these situations we adopt the following heuristics: First, since we know that the vehicle always is parallel to the floor, we mark a measurement unreliable if it reports the pose deviating from this more than a pre-set threshold. Second, the vehicle has a limited acceleration, which means that the position cannot change too much between two measurements. If it reportedly changes more than a pre-set threshold, the measurement is marked unreliable.

4. Experimental Setup

This section describes the experiment that was devised to demonstrate the system navigating in a space that is representative of a worst-case factory environment.

4.1. Facility

The experiment was first performed in Boeing's Collaborative Autonomous Systems Lab (CASL), previously known as the Vehicle Swarm Technology Lab (VSTL) [13]. CASL is a 30 x 15 x 6 m indoor room that couples a motion capture system with the ability to control vehicles based on visual feedback. The CASL is shown in figure

Fig. 1: Boeing Collaborative Autonomous Systems Lab.

A motion capture system from Vicon is used to provide localization data in the form of position and attitude for all objects in the test volume with 6 degrees of freedom (DOF). The CASL motion capture system includes 44 cameras with 16 megapixel and 4 megapixel resolution variants running at a capture rate of up to 1000 FPS.

The system utilizes coordinated pulses of visible light reflecting from markers attached to objects of interest. The system combines the returns of the multiple cameras to triangulate the individual marker positions. Objects are defined as patterns of multiple markers that the software then tracks.

The output is high rate, high accuracy measurement of position and attitude. Position accuracy is sub-millimeter and angular accuracy is sub-degree. Data for multiple objects is provided at 100 FPS with latency of approximately 10 milliseconds.

The system is easily calibrated using a wand with multiple markers at predefined spacing. A user moves the calibration wand in the space for few minutes to allow the system to back-calculate the camera positions. Then, another L-shaped frame also with markers is used to set the position and orientation of the measurement origin.

4.2. Robot Testbed

For this experiment, we employed a Kuka youBot as the testbed. The youBot is a small educational robot that is intended to be representative of larger mobile industrial robots. It is equipped with Mecanum wheels, allowing for independent motion in 3 axes: longitudinal, lateral, and yaw. A robotic arm with 5 joints and a 2-finger gripper is attached to the base.

The youBot base contains a battery, motor controllers, and a fully-programmable embedded PC. On the embedded PC, we implemented an inner-loop controller for wheel rotation given longitudinal, lateral, and yaw velocity commands. Velocity commands were received from an outer-loop computer running on an off-board computer via a serial radio.

Odometry measurements were transmitted over the same serial radio. A controller for the arm was also implemented.

4.3. PTAM-Based Monocular SLAM

To perform monocular SLAM, a custom computer vision package was mounted to the youBot. This package consisted of a high-speed monochrome camera (Point Grey FL3-U3-13Y3M-C) connected to a Microsoft Surface Pro via USB 3. This system performed monocular SLAM in an image stream with resolution of 640 x 480 pixels at 100 frames per second. The results of the monocular SLAM were transmitted to the off-board computer running the fusion algorithm via a serial radio. Figure 2 shows the youBot with the custom computer vision package and serial radios mounted.

Fig. 2: Kuka youBot with our custom vision package.

The algorithm that we employed was a heavily-modified version of the popular parallel tracking and mapping (PTAM) algorithm, published in 2007 by George Klein and David Murray of the University of Oxford [5]. Originally intended for augmented reality, it has become popular for SLAM due to its open source and ability to run on light hardware [6]. One of PTAM's main advancements was that it split the localization and mapping tasks into separate CPU threads. This allowed those tasks to be run in parallel but at different rates. Specifically, it is desired to update the localization with every incoming image, while expanding the map is much less frequent. This allows for some optimization (bundle adjustment) to be performed on the map in between map expansions.

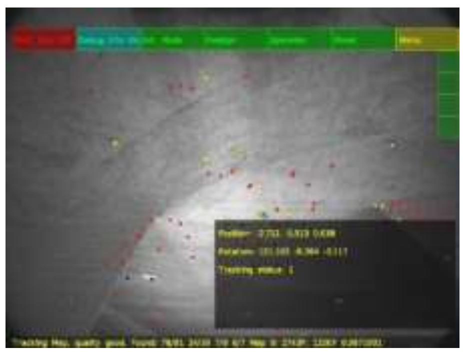

Fig. 3: Sample output from PTAM as the robot enters an enclosure.

In previous work, we had made several modifications to the PTAM algorithm to make it more suitable for robot localization using a high-speed image stream [7]. Since augmented reality is not the desired application, we split the rendering and localization into separate threads. We also added another thread for communication with a controller. We enacted adaptive thresholding for the feature detector to deal with changes in environments and modified the heuristics for expanding the map. Finally, we implemented an automatic (re)initialization method and the optional ability to set the origin according a fiducial marker. Figure 3 shows a sample screenshot of the modified PTAM software as the robot navigates into an enclosure.

4.4. Off-Board Computing

While the monocular SLAM and inner-loop controller were processed on-board the youBot, the other elements of navigation, control, and sensor fusion were processed in a modular fashion on off-board computers. These processes include: the motion capture (Vicon) algorithm and transmitter, the outer-loop vehicle controller, the sensor fusion algorithm and user interface, and the navigation interface.

4.5. Experimental Plan

Several navigation obstacles were constructed in the Collaborative Systems Lab in order to represent a worst-case factory environment. These obstacles included a pair of enclosures, one illuminated and one unlit, and a pair of wing sections. The experimental course is shown in figure 4.

Fig. 4: Experimental course.

The first task in the experiment was for the robot to navigate into the illuminated enclosure, make a 90-degree turn and then exit. The illuminated enclosure was designed to obstruct the view of the motion capture system, thereby forcing the system to recognize the failure and switch to an alternate sensor.

The second task was to navigate into the unlit enclosure, again make a 90-degree turn and exit. The unlit enclosure was designed to both obstruct the motion capture system and to be too dark for monocular SLAM. This should force the system to recognize both failures and switch to odometry.

The final task involved navigating under a pair of wing sections. This obstacle was most similar to the kind of obstacle that might be found in a factory environment. This kind of obstacle may or may not obstruct the view of the motion capture system, and may or may not be suitable for dependable tracking with monocular SLAM. It is therefore a good test of the system's robustness.

Fig. 5: The robot entering the illuminated enclosure, exiting the illuminated enclosure, entering the unlit enclosure, and traversing under the pair of wing sections.

The robot was commanded to perform each task in one continuous maneuver without stopping. Sample images of the robot performing each task are shown in figure 5.

5. Results and Conclusion

The robot was able to repeatably complete the course successfully, in spite of the obstacles that were described in the previous section.

5.1. Summary of Results

To facilitate visualization of the fusion algorithm, we developed a graphical interface. The interface shows a grid, representing the experimental area. Each grid is a meter square. An avatar of the robot appears at the location reported by each sensor, post-alignment. Each sensor is assigned a unique color. Overlapping areas of each avatar darken each other, such that the combination of all 3 colors forms black. Each avatar also leaves a trail indicating previous measurements. To the left of the grid are the current readings from each sensor. A black box around the reading indicates the preferred sensor, whose measurements are currently being used for navigation. Readings appear grayed out when a sensor is not producing new measurements. Sample screen captures from the graphical interface during the experiment are displayed in succession in figure 6.

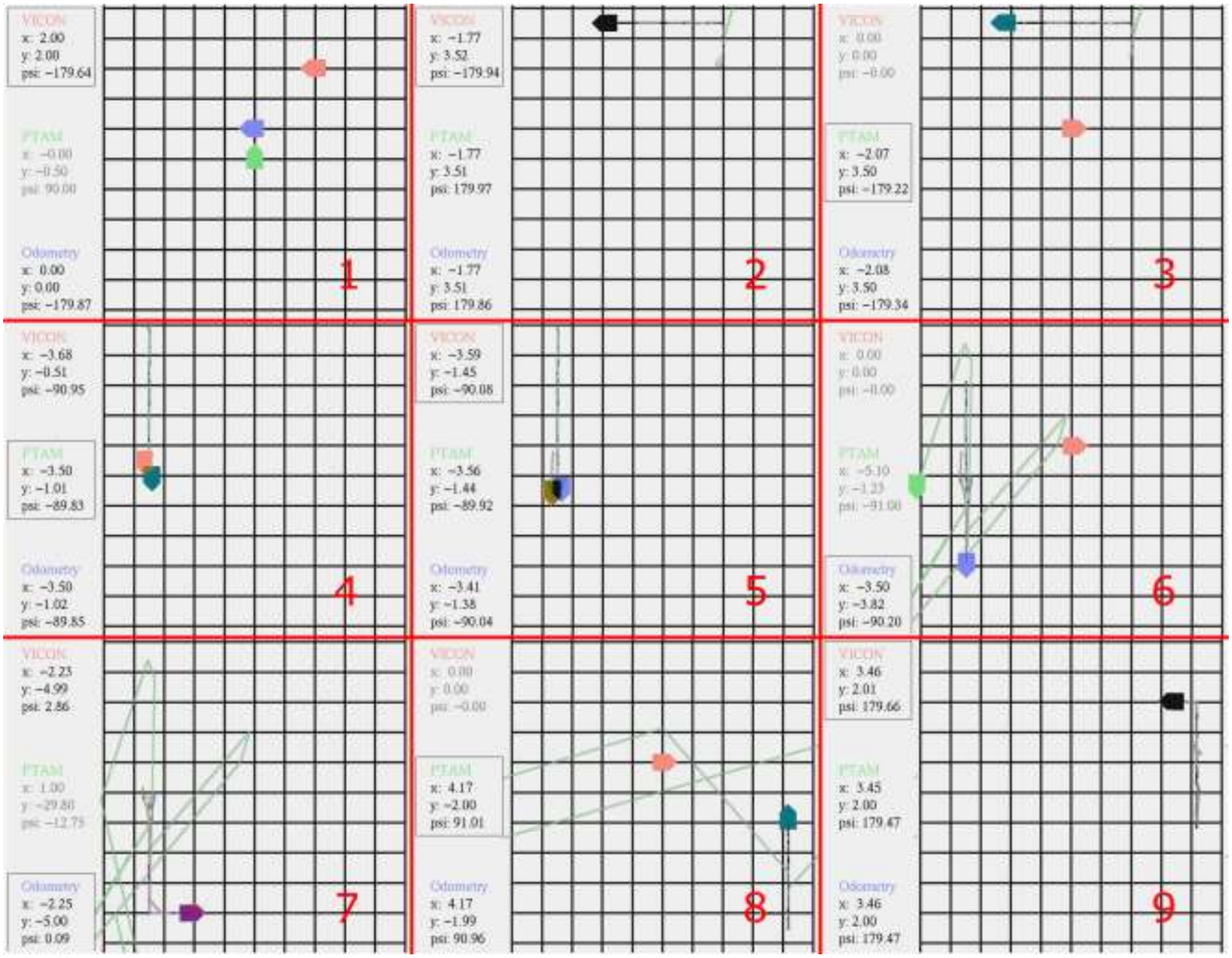

Fig. 6: User interface screen captures.

Screen capture 1 shows the state of the system prior to beginning the navigation task. In this case, both Vicon and odometry are producing measurements, though the relation between their reference frames cannot yet be estimated. Monocular SLAM requires motion to be initialized, so it is not yet producing measurements.

Screen capture 2 shows that the robot has advanced toward the illuminated enclosure. All 3 sensors are producing measurements and are well aligned. Vicon is the preferred sensor because its accuracy and precision are higher than the other 2 sensors.

Screen capture 3 occurs when the robot has entered the illuminated enclosure. The robot is no longer visible to the motion capture cameras and the system has detected that those measurements are not reliable. The system has seamlessly transitioned to monocular SLAM. Odometry is also available, but its accuracy is lower than that of monocular SLAM. Odometry is now being aligned again monocular SLAM.

Screen capture 4 occurs when the robot has exited the illuminated enclosure. We see that Vicon is producing measurements again and that some error has accumulated in the other measurements. Vicon has not been determined to be reliable yet. In screen capture 5, it can be seen that the software has determined Vicon to be reliable and has switched. Both monocular SLAM and odometry will now be realigned to Vicon.

Screen capture 6 occurs after the robot has entered the unlit enclosure. The robot is no longer visible to the motion capture cameras and monocular SLAM has produced some unreliable measurements before ceasing altogether. However, the system has successfully detected these faults and has seamlessly switched to odometry. Screen capture 7 shows that the robot successfully exits the unlit enclosure, where it again becomes visible to the motion capture cameras. Notice that the odometry estimate is still aligned quite well with Vicon. The system switches back to Vicon and monocular SLAM is reinitialized shortly after this.

Screen capture 8 occurs when the robot is traversing under the pair of wing sections. Here the motion capture system occasionally loses sight of the robot and monocular SLAM occasionally provides unreliable measurements. The system is able to cope with these faults and successfully completes the course, as shown in screen capture 9.

5.2. Conclusion

In this paper we have reported a successful demonstration of a robust navigation system for a factory environment. In particular, we combined a motion capture system, monocular SLAM, and odometry to form robust position and rotation measurements. We demonstrated an algorithm to align the reference frames of each sensor, assess the reliability of the measurements reported by each sensor, and perform switching when necessary. With this system our robot was able to repeatably navigate through an experimental course at Boeing’s Collaborative Autonomous Systems Lab. This course was representative of a worst-case factory environment.

6. References

[1] Vicon. url : http://www.vicon.com (visited on 04/03/2015).

[2] Ducard, Guillaume, and Raffaello D'Andrea. "Autonomous quadrotor flight using a vision system and accommodating frames misalignment." *IEEE international symposium on Industrial embedded systems, 2009. SIES'09*. IEEE, 2009.

[3] Mellinger, Daniel, Nathan Michael, and Vijay Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors." *The International Journal of Robotics Research* (2012): 0278364911434236.
http://dx.doi.org/10.1177/0278364911434236

[4] Raabe, Chris, et al. "Aggressive navigation using high-speed natural feature point tracking." *IEEE Aerospace Conference, 2014*. IEEE, 2014.
http://dx.doi.org/10.1109/aero.2014.6836340

[5] Klein, Georg, and David Murray. "Parallel tracking and mapping for small AR workspaces." *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007. ISMAR 2007*. IEEE, 2007.
http://dx.doi.org/10.1109/ismar.2007.4538852

[6] Klein, Georg, and David Murray. "Parallel tracking and mapping on a camera phone." *8th IEEE International Symposium on Mixed and Augmented Reality, 2009. ISMAR 2009*. IEEE, 2009.
http://dx.doi.org/10.1109/ismar.2009.5336495

[7] Raabe, Chris, et al. "Efficient alignment of position measurements with differing reference frames." *Asia-Pacific International Symposium on Aerospace Technology, 2014. APISAT, 2014.*