Fusion of Road Surface using RGB-D Sensors

Ansir Ilyas, Muhammad Abdullah

[1] Department of Electrical Engineering
SBA School of Science & Engineering – LUMS, Pakistan

{14060038,14060024}@lums.edu.pk

1. Introduction

Roads undergo rapid deterioration due to various economic, social and natural reasons. The uncertainty and unavailability of road safety information has become a major issue in reliably transporting goods, movement of heavy machinery, transport of people and materials to/from disaster areas and in even giving plain guarantees for safe traversability of a road that is good on paper. Given a starting location and destination, a desktop or mobile mapping service provides a path that any typical vehicle might take. These vehicles does not return any exact answer for any particular type of vehicle. Nor do they report the relative difficulties that a driver may face while traversing the path. In this report we try to use use vision base techniques to detect the negative obstacles of particular road patch.

Intelligent navigation and path planning requires some sensors which provide information of surroundings. Based on this information decision is taken or a parameter is updated like building a map, localization or planning a path. Since, each sensor has some limitations and inherent uncertainty in sensor readings which may accumulate over time and resulting in erroneous decisions. Sensor fusion is one of the techniques to reduce this error in the readings and overcome the limitation of sensors by utilizing multiple sensors. These sensors can be of same type or different sensors can also be used to complement the information of each other. Multisensor data fusion is a process to combine the information from number of different sensors to provide a robust and complete information about the environment.

In this project, data fusion is used to detect obstacles on the road and get their accurate location. We will also analyze the accuracy of different sensors. The different sensors we will use are Kinect vision sensor, laser hokuyo sensor to detect the obstacle and GPS and IMU modules to measure their position.

2. Related Work

Several approaches have been presented for 3D environment reconstruction, using different sensors (cameras, stereo cameras, multiple 2D LRF, 3D LRF, and combinations of them). For example, (Diebel et al., 2004) use active stereo vision for building a 3D metric map of the environment, (Thrun et al., 2004, Fruh, 2004) use two orthogonal 2D LRF to build 3D maps of indoor and outdoor environments, while (Nuchter et al., 2005) use a 2D LRF mounted on a tilt unit that is able to acquire a very precise 3D scan of the environment with a relative cheap sensor, but it requires a higher acquisition time due to the rotation of the laser. approaches based on feature extraction and computer vision techniques have been proposed (e.g., MonoSLAM (Davison et al., 2007)), providing for 3D feature-based maps. Outdoor mapping has also been investigated. For example, in

(Konolige et al., 2006) the use of stereo vision and visual odometry has been proposed for long distances outdoor navigation of a mobile robot. All these approaches are focused on building metric or feature based maps, either considering relative small environments to map or focussing on the navigation capabilities of an autonomous platform.[1]

3. Approach

Previously traversability index is calculated using Kinect only. It has a limited range and fails in presence of day light due to interference of infrared rays from sun. To cater this limitation we use long range laser sensor with Kinect and Inertial Measurement Unit (IMU) incrementally calculated vehicle position and orientation.

As mentioned earlier, we are using Kinect XBOX 360, laser scanner and IMU for getting information of road surface. To acquire data from these sensors we need to configure and simultaneously run all these sensors. Kinect and laser scanner are directly configurable with ROS. However, we need an IMU that could also directly configured with ROS. Initially we decided to use IMU of our mobile device which transmits data on UDP stream in CSV format. We wrote a code for Socket programming that could receive UDP data on a port. Decoding this data and populating the */imu* topic was cumbersome so we decided to use ardrone to IMU data from it.

3.1. Hardware Platform

The testing was done on the platform shown in Fig. 1. For initial testing we recorded a bagfile of the corridor of electrical engineering department by placing small obstacles as shown below in Fig. 2.

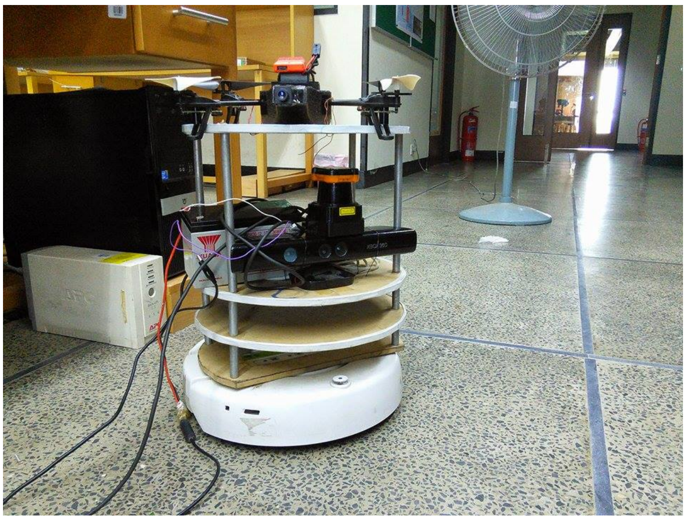


Figura 1. Initial platform used

There was additional overhead of configuring iRobot in this assembly. Moreover, we couldn't tilt Kinect and laser sensor at an angle to obtain the view of the floor. Figure 3 assembly was developed to house all the sensors with a freedom to tilt the sensors at an agle of our requirement.

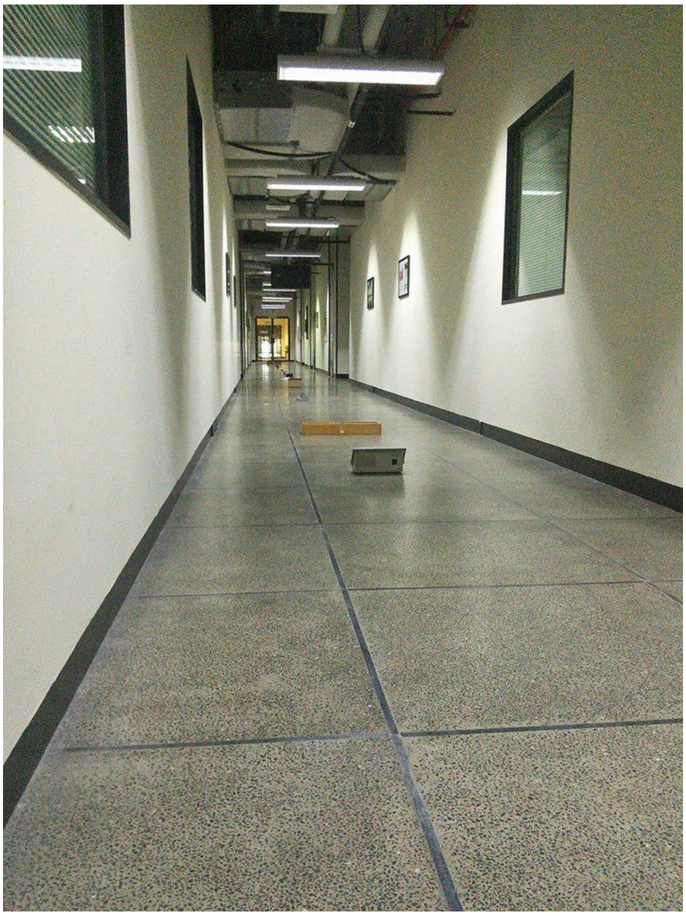


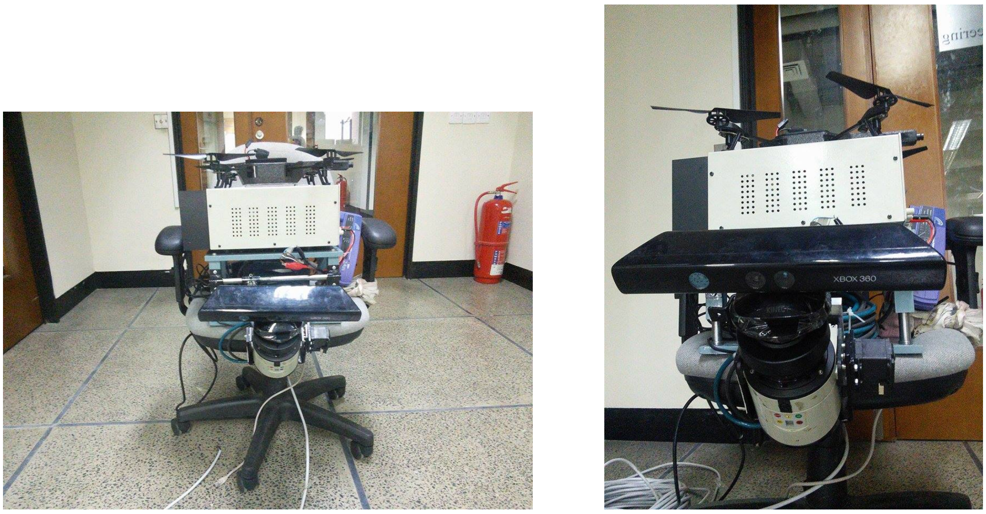
Figura 2. Bag recorded for test by placing obstacles in corridor

3.2. Sensor Configuration

Next step is to configure these sensors. Laser hokuyo was configured by running *hokuyo_node*. Similarly, kinect was configured by executing openni.launch file from the package *openni_launch*. These two sensors provide us the depth information but in different formats. Laser scanner provides accurate 2D depth information and Kinect gives 3D point cloud. Ardrone was also configured to get imu data.

3.3. Recording Bagfile

For offline processing we need to record data in a bagfile. When the data was recorded using the *-a* option it exceeded the limit and initial data started to drop. After little exploration it was observed that setting a parameter *depth_registration* false during the recording and changing it to true when playing the bag processes the generates the a point cloud offline. *camera/depth/image_raw* and *camera/depth/camera_info* are the topics required for XYZ-only point clouds. However, for RGB point clouds *camera/depth_registered/image_raw*, *camera/depth_registered/camera_info*, *camera/rgb/image_raw* and *camera/rgb/camera_info* are the required topics. We have recorded two different bags to process point clouds in both formats.



(a) (b)

Figura 3. New platform with different sensors used

3.4. Publishing Static Transforms

Both camera and laser are in different frames. /camera_link is the static frame for the kinect and /laser is the static frame for laser. All these frames need to be transformed to base frame which is base_stabilized in our case. Since our aim is to detect negative obstacles on the road surface using RGB-D sensor, therefore to acquire road surface data, we mount laser sensor at some particular degree on mobile robot assembly. After measuring this static transform we have viewed this data in Rviz which shows that both the sensors are detecting data at the same time. Static Transform published is shown in Fig. 4. The result of view_frames is showin Fig. 5

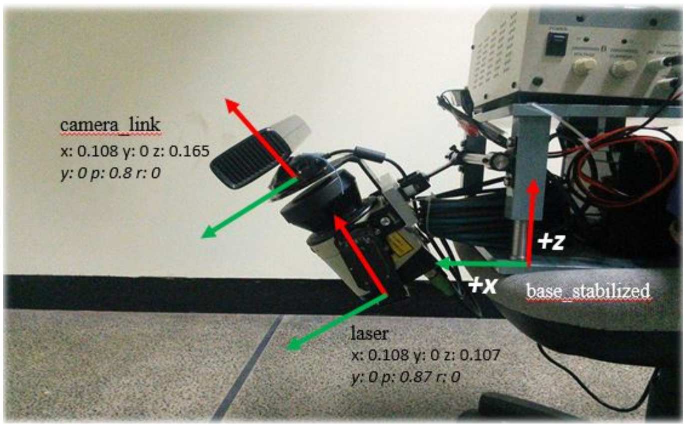


Figura 4. Transformations parameters to base frame

3.5. Hector Mapping

Hector mapping is a technique to implement SLAM only using laser scan. Unlike gmapping it does not require odometry data. We have created a hector map using the recorded laser scan data from the bagfile. Hector mapping can accomodate the information roll/pitch of the sensor to accurately map the environment. So, we have executed

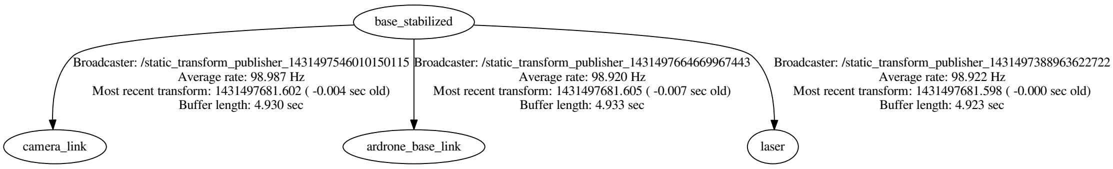


Figura 5. Transformations parameters to base frame

imu_attitude_to_tf node which takes care of it. Transformation required for hector mapping are shown in Fig. 6 and view_frames result is shown in Fig. 7

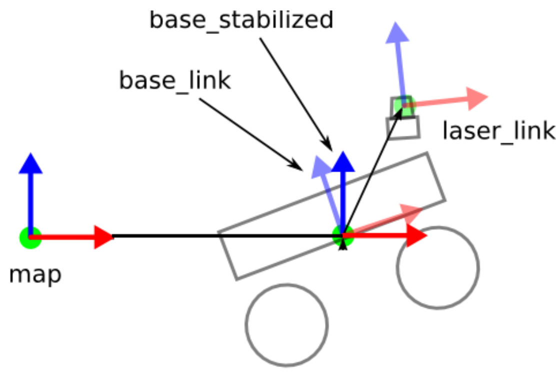


Figura 6. Transformations required for hector mapping

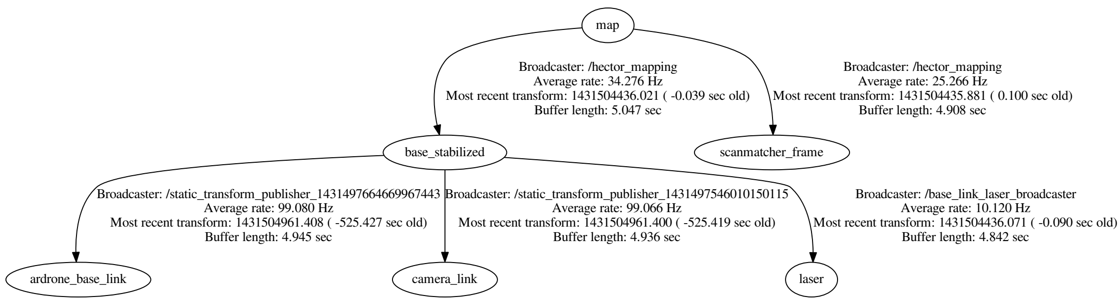


Figura 7. Transformations required for hector mapping

4. Fusion of Data

Data fusion of laser scanner and kinect would greatly improve the quality of our data and reduce the amount of outliers. After fusion data is used to detect negative obstacles on the road surface and this information can be translated to road traversibility index (RTI). We use a map created by laser to detect the obstacles. For a transparent object laser scan gives incorrect scan. In such scenario, information from Kinect can be used to account for this error.

4.1. Fusion using Hector map

We acquire the data of particular patch of road from Kinect and laser. Kinect gives us 3D point cloud and laser scan give us 2D line. In order to fuse Kinect point cloud and laser scan line first we need some 2D map. There are several techniques available for mapping such as gmapping and hector mapping. We have created a map using hector map as we currently don't have the information of odometry and hector map is built using laser scanner data only. Next step after acquiring 2D map and 3D point cloud is to find and match corresponding points align the information of these two sensors.

The requirement for generating hector map is that laser scanner should be parallel to base frame. However, the requirement of our project is to tilt this sensor at a certain angle from the base frame due to which hector map could not be generated successfully. So, a different method for fusion was required.

4.2. Fusion using Line Extraction

In this case, after acquiring the 3D point cloud from Kinect and laser scanner we try to extract the line at same depth as the laser scanner looking towards the road surface as shown in Fig. 8. Then we fuse these lines to improve the quality of our data and reduce outliers.

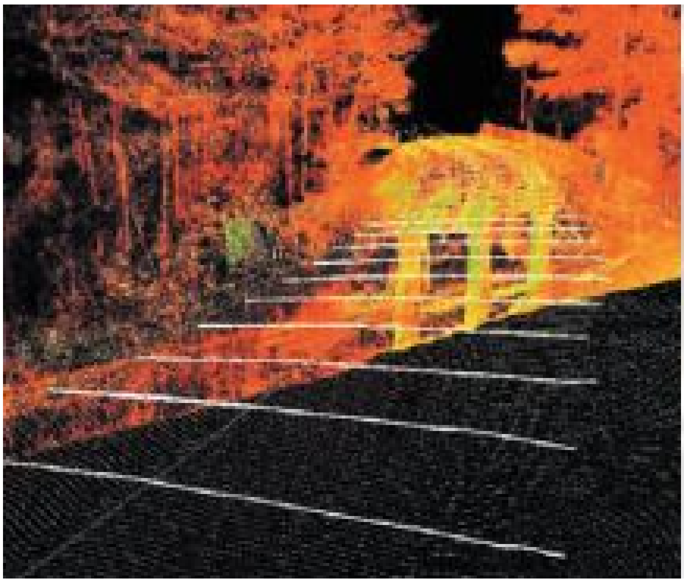


Figura 8. Line extraction from point cloud

5. Experiment & Analysis

After initializing all the setup we started Rviz to visualize the data of laser scan. Since all the transformation have been published, we need to specify the based_stabilized frame in fixed frame in Rviz as shown in Fig. 9

Looking at the point cloud in Fig. 10 we can observe the obstacles.

By visualizing both the sensors at the same time in Rviz we confirm that the obstacle appears in the frame of camera and sensor occur at same time as shown in Fig. 11

Hector map visualizations is also given in the Fig. 12 below

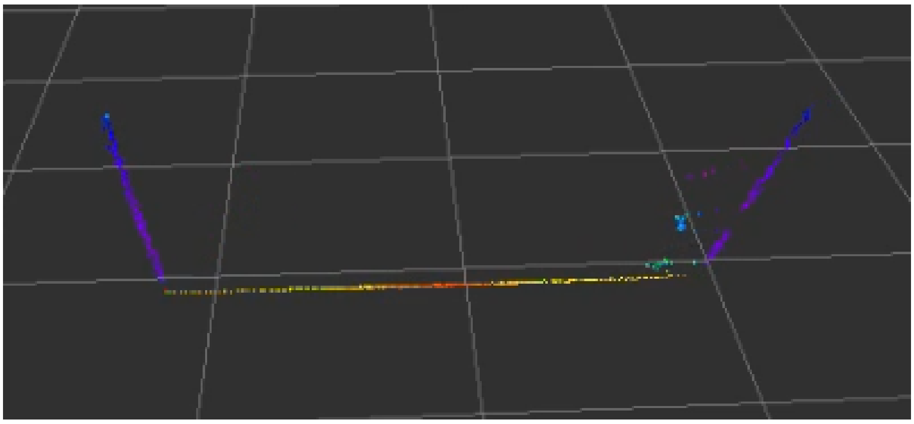


Figura 9. Laser scan after transformation

Figura 10. RGB point cloud after transformation

6. Conclusions & Future Work

We have performed an initial setup for recording data and transforming data to the same frame. Now, using different techniques of line extraction and plane extraction from 2d map and 3d point cloud could be matched to construct a complete 3D map of the envioronment.

7. References

[1] Building 3D maps with semantic elements integrating 2D laser, stereo vision and IMU on a mobile robot, Iocchi and Pellegrini

[2] Vehicle specific traversibilty analysis of pathways, Muhammad Mudassir Khan

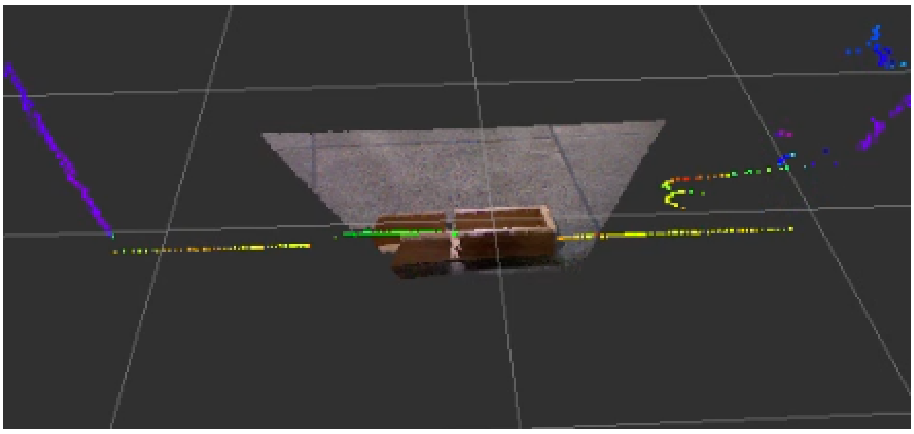


Figura 11. Fusing information from both sensors

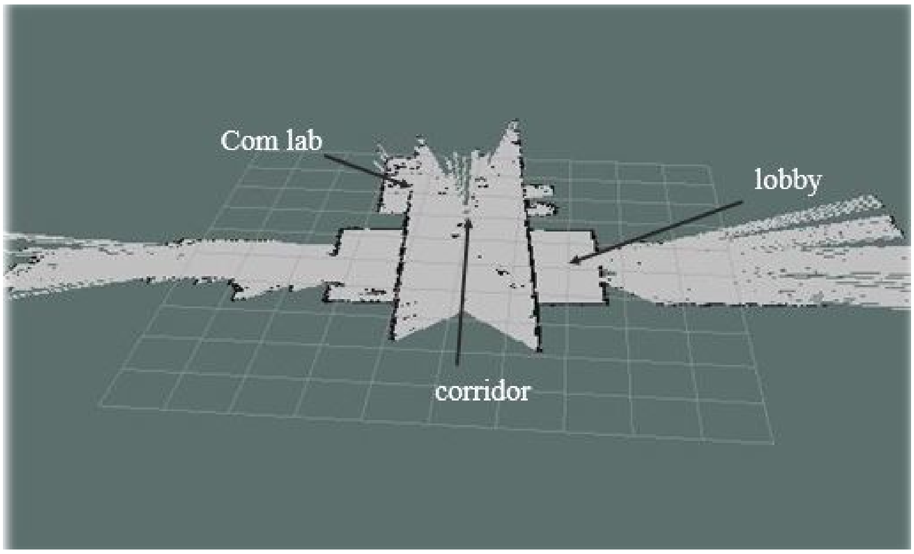


Figura 12. Hector map of corridor