Modelling of Terrain Surfaces Using Aerial Radar Mapping

David Murdoch — University of Waterloo, ON, Canada
Eric Ye — University of Waterloo, ON, Canada
Dr. George Shaker — University of Waterloo, ON, Canada
Dr. William Melek — University of Waterloo, ON, Canada

Abstract

Aerial scene mapping is often done via visual methods, where many 2D images are combined together to create 3D maps. There are several disadvantages to this approach, however, including weather interference, inconsistent or nonexistent lighting, or fast-moving objects, which often appear as either noise or a blur on the created map. Radar technology is mostly used in automotive applications, like lane keeping and adaptive cruise control. It can also be used, however, for aerial mapping of scenes, by mounting the radar to a drone. The radar can then be used to generate a point cloud, that can either replace or complement point clouds and maps from other sensors. We present an approach for generating a point cloud map using aerial radar. We then present an algorithm for aggregation of points based on pose, and a method to create accurate meshes over mapped terrain, to facilitate path planning by ground-based robots.

1. Introduction

In aerial robotics, when mapping techniques are necessary, typically camera-based techniques are used, such as SVO[1] or DSO[2]. However, these techniques face significant difficulties, including changing weather conditions like rain or snow, lighting condition changes, or just relatively uniform looking terrain, like sand, snow, or trees. Radar, however, is unaffected by weather and illumination changes – depending on the frequency of operation, allowing it to perform optimally in a wider range of conditions and environments. The radar can then be used as an alternate method of imaging and mapping.

Aerial robots, in addition to operating individually, can also be used as support vehicles for ground vehicles. In particular, maps created by an aerial vehicle can be used by a ground vehicle for planning paths. After mapping, processing and segmentation of the point cloud are performed, to retrieve the point cloud describing the ground and ground level obstructions. A surface can then be fit to this point cloud, to describe the terrain for use in other applications.

Section 2 of this paper briefly describes existing UAV mapping systems. Section 3 gives an overview of the algorithms used, beginning with the radar techniques for creating the point cloud, then the algorithm for aggregating points and transforming them. Section 4 describes the mesh creation process. Section 5 presents the results and discusses conclusions.

2. Background

Monocular camera has been shown to be effective at mapping indoor and outdoor spaces, as well as providing pose estimation and depth estimation using SVO and REMODE [3], by moving a camera over a scene to be mapped. However, this has some disadvantages. There is significantly lower accuracy when there are fewer visible features, such as in a snowy area or in a grassy uniform field, and the scene must be well lit and clear of visual obstructions, including conditions like fog or night time. Other efforts have attempted the use of radar for mapping before, using Synthetic Aperture Radar (SAR) to perform mapping of outdoor scenes, which was demonstrated by mapping both a populated town, and rural areas in [4].

3. System Description

3.1. Radar System

We use frequency modulated continuous wave (FMCW) [5] principle to perform radar to target measurements. The utilized low-cost radar has 3 Tx antennas and 4 Rx antennas, a Tx gain of 12 dBm, and operates at 79GHz with 4GHz for bandwidth. We set the Rx chain gain to be 30 dB using an LNA. The radar ADC is configured to output 16-bit complex samples. Unless otherwise noted, all processing takes place on the radar chip itself, as it includes an ARM Cortex R4F microcontroller, with radar hardware acceleration. The frame and chirp parameters used for this experiment are summarised in Table 1. After processing is complete, detected objects are outputted from the radar using a Universal Asynchronous Receiver/Transmitter (UART) connection to create a point cloud.

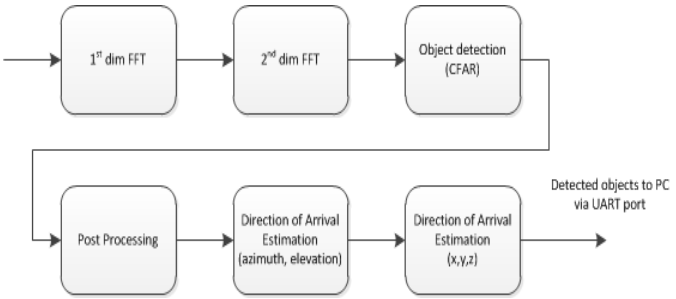

Fig 1. A block diagram of the radar system.

The radar is attached to a drone, which is equipped with GPS and IMU sensors for pose estimation.

3.2. Object Detection

After taking the 1st and 2nd dimension FFT, we use the resulting range-Doppler matrix to perform simple constant false alarm rate (CFAR) detection of objects. We use an 8-sample window and 4 guard samples for (CFAR) detection. Objects are identified by range and doppler index from CFAR, as well as the calculated noise energy in the cell for CFAR. In order to conserve both throughput over the UART and processing power, we also use a peak-grouping scheme, in which for each peak, neighboring cells in which objects are also detected, but are less than the peak, are discarded.

After post processing was completed, another 2 complex FFTs were performed on the azimuth and elevation antenna array data, along with a log-magnitude FFT on the azimuth antenna data. These results are then used to perform direction of arrival

estimation, to convert points with detected objects into Cartesian coordinates [6]. First, the range of the point to be converted from the radar is calculated using (1), in which k_r is the range index from CFAR detection, f_s is the sampling frequency in Hz, *N* is the 1D FFT size, and f_c is the frequency slope in Hz/sec, as defined in Table 1.

$$R = k_r \frac{cf_s}{2 f_c N} \quad (1)$$

Next, we define the angle w_x using (2), where k is the range index of the point to be estimated in the log magnitude FFT.

$$w_x = \frac{2\pi k}{N} \quad (2)$$

Using w_x, we can then compute a similar angle w_z as in (3), with P_1 and P_2, respectively, are being defined as the complex numbers corresponding to the point to be estimated in the azimuth and elevation array FFT.

$$w_z = angle(P_1 P_2^* e^{-2jw_x}) \quad (3)$$

Next, w_x and w_z can be converted to coordinates *x*,*y*, and *z* with (4) and (5). Given that R is known, we can solve for *y* using (6) to complete our Cartesian coordinates.

$$x = R\frac{w_x}{\pi} \quad (4)$$

$$z = R\frac{w_z}{\pi} \quad (5)$$

$$y = \sqrt{R^2 - x^2 - z^2} \quad (6)$$

These Cartesian coordinates are then sent via UART to a receiver to perform mapping.

3.4. Post Processing

The point cloud was improved by filtering out objects to reduce clutter and "phantom" objects, caused by reflections in the scene or leakage between the transmit and receive antennae. This is performed by removing points that are closer than a certain threshold (50cm) to the radar while the radar is in the air and objects are known to not be in that vicinity. In addition, if the drone is estimated to be a certain distance *x* meters above the ground, points more than 1.5*x* meters away from the radar can be ignored. Finally, since the accuracy of the angle-of-arrival estimation is lower at higher angles, objects at a point greater than a certain angle from the principal axis (the axis that extends out of the radar downwards to the ground) are removed. In this case, points more than 25° from the principal axis are removed.

Finally, to remove spontaneous noise from the point cloud, objects can be clustered and points that do not belong to a cluster are removed from the image. The DBSCAN algorithm [7] is used for this task.

3.5. Point Integration

In order to create a map of the entire scanned region, radar point clouds taken during flight must be compensated by the position of the drone at the moment the radar captured the point cloud. To do this, the *n* points P_i in the point cloud are compensated according to the pose of the drone.

The pose of the drone is given by two vectors: *T* (the translation with elements *Tx, Ty, Tz),* and *q* (the rotation of the drone in quaternion format). *q* is a vector of four elements which represent the orientation of the drone [6]:

$$q = q_0 + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 \quad (7)$$

A rotation matrix can be formed from this quaternion:

$$R_q = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_2-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2q_1^2-2q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2q_1^2-2q_2^2 \end{bmatrix} \quad (8)$$

A point $P_i(x,y,z)$ in drone coordinates can be converted to world coordinates by being rotated by the quaternion and then adding the translation. Explicitly, this is shown in the equation:

$$P' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_q \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (9)$$

Where $P_i'(x',y',z')$ is the compensated point in world coordinates.

By converting point clouds from the radar images to world coordinates and integrating them over the flight, a point cloud map of the environment and terrain can be made.

4. Mesh Creation

After the aggregated point cloud is complete, the point cloud itself must be processed to remove noise and smooth the point cloud. First, outliers are removed from the data by using sparse outlier removal [8], where the mean μ and standard deviation σ of nearest neighbor distances are calculated, and any points which fall outside of $\mu \pm \alpha\sigma$ are discarded. Next, the point cloud is down sampled and a grid average is taken, allowing for some preliminary smoothing [9].

Next, the point cloud must be segmented to avoid irregularities when creating the mesh. As can be seen in Figure 3 in the results, obstacles with greater heights, or suspended objects such as lampposts, manifest in the point cloud as clusters separated from the ground. To eliminate these before generating a surface, we use DBSCAN [7] to cluster the point cloud. The ground is defined as the largest cluster with centroid within 0.2m of ground level, as measured from the sensors of the drone. All other clusters are removed, as dealing with them is beyond the scope of this paper.

After overhanging objects have been removed, then further grid averaging is used, in order to counteract more angle-of-arrival estimation error. Piecewise linear interpolation is then used to form a function which can approximate the surface as measured by the radar.

5. Results

To test the algorithm, a drone with radar attached was flown above a parking lot with two cars and a lamppost, in order to test the mesh generation algorithm. The test area is shown in Figure 2, while the resulting aggregated point cloud is shown in Figure 3.

The resulting point cloud also includes a third vehicle, which parked in the parking lot during the test.

Fig 2. A view of the area where the flight test was conducted.

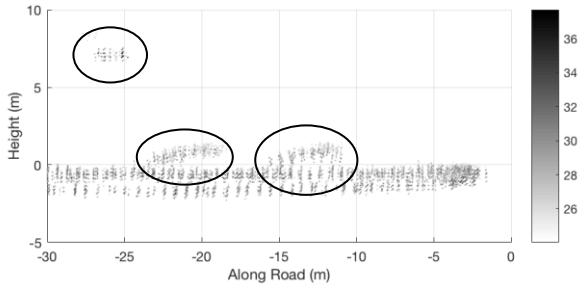

Fig 3. A side profile of the results from the flight test. The 2 cars and the lamppost are circled.

A processed point cloud and the resulting surface are shown in Fig 4 and Fig 5, respectively.

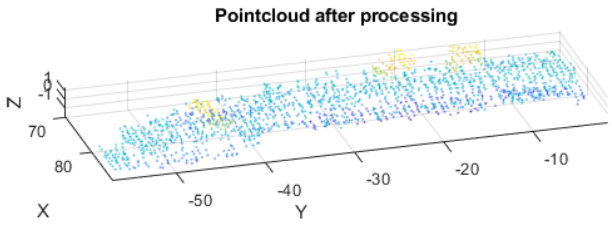

Fig 4. Processed point cloud from the flight test.

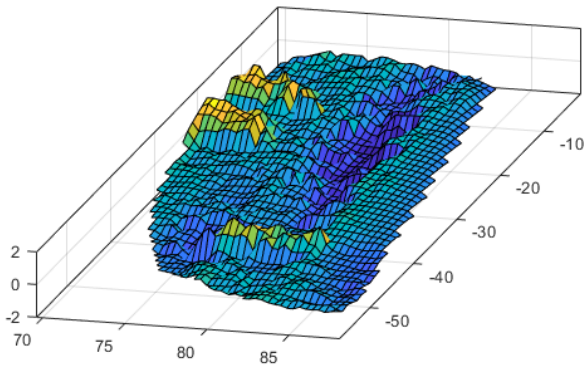

Fig 5. The surface post processing.

In this surface, the cars can still be seen as relevant obstacles, with the orange topped protrusions from the mesh being the two parked cars and one car that drove through the parking lot while testing was underway.

6. Conclusions

Data from the test indicates that surfaces can be modelled with reasonable accuracy using radar data from a drone. This technology is low cost, low weight, relatively low power, and accurate. This technology could be extremely useful for future surveying efforts in autonomous aerial robotics, including for any sort of collaborative effort.

There are many areas that can improve this algorithm. The signal parameters for the radar could be optimized to increase accuracy, the angle-of-arrival estimation can be optimized to reduce error inherent in the approach, and a spatial Gaussian filter could be implemented to smooth data instead of using grid averaging.

Acknowledgement

We would like to thank Dylan Gareau, a graduate student at the University of Waterloo, for helping with the flight tests, providing insight into drone architecture and camera models, and providing instruction in how to fly the drone.

References

[1] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014.

[2] C. Forster, Z. Zhang, M. Gassner, M. Werlberger and D. Scaramuzza, "SVO: Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249-265, 2017.

[3] M. Pizzoli, C. Forster and D. Scaramuzza, "REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014.

[4] P. Kaniewski, C. Lesnik, P. Serafin and M. Łabowski, "Chosen results of flight tests of WATSAR system," in *International Radar Symposium (IRS)*, Krakow, Poland, 2016.

[5] M. Skolnik, *Introduction to radar systems*, Boston [u,a]: McGraw-Hill, 2007.

[6] *MMWave SDK: Millimeter Wave demo.* Dallas, Texas, Texas Instruments Incorporated, 2017.

[7] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, USA, 1996.

[8] Rusu, R. B., Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. “Towards 3D Point Cloud Based Object Maps for Household Environments”. Robotics and Autonomous Systems Journal. 2008.

[9] Pomerleau, F., F. Colas, R. Siegwart, and S. Magnenat. “Comparing ICP variants on real-world data sets.” Autonomous Robots. Vol. 34, Issue 3, April 2013, pp. 133–148.