

592 Proposal: SAR Imaging via Back-Projection

John Kesler
jckesle2@ncsu.edu

I. TEAM

The team consists of myself, John Kesler – I’m an EE Master’s student (3rd semester) specializing in mmWave circuits. I graduated with a Bachelor’s in both EE and CPE in 2024. I have over 10y of programming experience (hobbyist, professionally at a software company, and academically as a researcher), with most of my time being split between Python, C, and Matlab/Octave.

The bulk of this project can be done using fairly simple Python code, but can benefit from GPU or parallel acceleration. For this part, I have some experience with Tensorflor (and sufficient personally owned GPU hardware) that I can leverage.

II. PROJECT

At a high level, the core goal of the project is **implementing a SAR radar imaging algorithm with an autofocus/motion-correction algorithm**. This goal is motivated by a personal interest in drone-borne SAR imaging.

A. Abstract (Motivation and Scope)

One of the key challenges to resolve when using a mobile platform for SAR imaging is dealing with the variability and lack of direct control over the radar’s trajectory. Using a drone as an example, it’s easy to command autopilot software to just “go to a point at a set velocity,” but without perfect atmospheric and mechanical conditions, the platform’s path will have some errors introduced.

Considering the time and resource constraints involved with this project, the scope will first, and primarily, be focused on collecting data from a SAR configuration and resolving an image. To achieve this first goal, a basic ground-based SAR algorithm such as the Omega-K algorithm [1] can be implemented. This algorithm (ultimately an improvement to back-projection) is easily parallelizable, and should be an interesting programming challenge. This algorithm, however, has an inbuilt requirement that the path velocity is well-controlled, so large deviations (relative to a wavelength) can lead to defocusing in the final image.

The second part of this project involves implementing an “autofocusing” algorithm – something that can take in a messy image, and phase-shift the data to correct for (potentially unknown) deviations from the ideal trajectory. There are a few established approaches to this problem, depending on the data collected and severity of the defocusing. I’d like to implement a more generic approach using basic machine learning via a minimum entropy autofocus algorithm as detailed in [2].

Ideally, meeting these two goals will create a good software base that can be used for future projects involving SAR imaging. A stretch goal, outside the scope of this class, of course, is attaching a radar to a drone and using it to perform imaging, such as in [3].

B. Objectives, Deliverables, Milestones

This project is broken into three phases. First, data will be collected in a controlled environment for SAR imaging, or an existing dataset will be found and preprocessed to match data output as seen from a lab radar. Secondly, the basic Omega-K processing algorithm will be implemented, and GPU-optimized using Tensorflor. Finally, some motion-correction or autofocus algorithm will be implemented.

If all goals are met and the software made is sufficiently robust, then a final round of data collection can be done in which a radar is carried for ground-based imaging outside a controlled environment.

The deliverables/milestones are as follows:

- 10-20/10-26: Radar setup, data collection
- 10-27/11-02: Basic imaging/BP implementation
- 11-03/11-09: GPU-accelerate algorithm
- 11-10/11-16: Implement autofocus
- 11-17/11-23: Any spillover/more (interesting) data collection
- 11-24/11-30: N/A – poster generation/documentation

Ultimately, a good outcome for this project is implementation of an algorithm that can generate 2d images of an arbitrary scene from raw data collected. The *best* outcome is improving the software to a point where, for example, a radar can be carried around campus and resolved images can be directly overlaid on satellite imagery with good results.

C. Feasibility Analysis

This section works backwards from the 77GHz TI radar’s parameters to see what’s possible in terms of imaging. The relevant parameters are described in Table I and are built on the assumption that the TI chip is on the AWR2243BOOST module (for antennas).

TABLE I
AWR2243 CHARACTERISTICS

Parameter	Value
Center Frequency	78.5 [GHz]
Bandwidth	5 [GHz]
Wavelength	3.8 [mm]
ADC Rate	45 [Msps]
Transmit Power	13 [dBm]
Rx NF	13 [dB]
Antenna gain	10 [dB]
Ramp rate	266 [MHz / us]
Pulse duration	18.7 [us]

With these values, we can plot a measurable RCS against range, with the assumption being that we need >1 (linear) SNR. This is a similar metric to NESZ in actual SAR systems, but in this application will let us determine a maximum (theoretical) range to which we can “see” targets – using known values of corner reflectors in the lab as a reference.

$$\sigma_{\min} = \frac{(4\pi)^3 R^4 k T_0 F \beta}{P_t G^2 \lambda^2} \quad (1)$$

$$\sigma_{\text{corner}} = \frac{4\pi a^4}{3\lambda^2} \approx 20 [\text{dBm}^2] \quad (2)$$

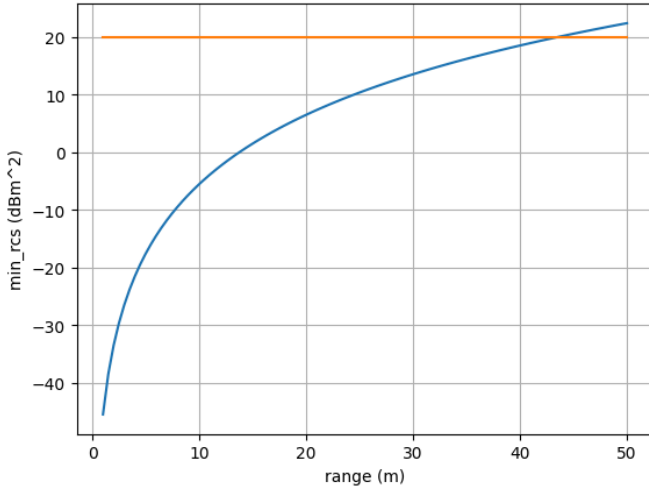


Fig. 1. Minimum detectable RCS at various ranges

This suggests that we can detect a corner reflector up to ≈ 40 m away using the TI radar.

The other constraint on sampling, in FMCW, is the unambiguous range, which is given in (3).

$$R_{\text{ua}} = \frac{c\tau}{2} = 2.8 [\text{km}] \quad (3)$$

This value is sufficiently high, as we will not expect to see anything beyond this value.

The final set of constraints to check is a comparison between the PRF of the device and the relation between ADC sampling frequency and target range.

The lower bound on the PRF is generally set by the speed of the SAR device compared to a quarter wavelength. At 78.5 GHz, a quarter wavelength is 0.95mm, suggesting that for a (high estimate) 3m/s velocity, the PRF needs to be at least $\frac{3}{0.95 \times 10^{-3}} = 3.2$ kHz. The TI radar does not list a minimum PRF, but 3kHz is quite low and likely can be achieved.

Finally, we can assume that the ADC sampling requirements are met by nature of the system being integrated and an evaluation board.

III. COLLABORATION PLAN

The tasks required for this project are described in Section II.B.

The bulk of this project will be completed by myself, however Josiah Neal (jrneal@ncsu.edu) has set up a rail system for SAR imaging and has confirmed a willingness to aid with any lab measurements using it. I'll coordinate with him over the next few weeks to either exchange data, or get time on the system.

A. Risk Analysis

The primary risks involved with this project, and mitigation plans are detailed below.

It's not possible to collect data from a lab radar/the rail: known good online SAR imaging datasets can be used instead. These datasets will also be used throughout the project to provide a ground truth for the imaging algorithms.

I don't have enough time to implement everything or a task stalls: the tasks within the project are well-partitioned and each provide a way to independently move forwards if something else fails. e.g., if I'm unable to implement the back-projection algorithm, then I can find an open source implementation and move forwards with implementing autofocus. Some deadtime is built into the schedule to allow for spillover for the sake of time management.

I have too much time, or the project ends up being too trivial: there are low-level processing improvements that I can make to the naive implementations of the algorithms. I have a personal interest in GPU parallelization, so more attention can be given to that aspect, if needed. Or, scope can be added to use a more powerful autofocus algorithm that takes in, as an example, motion data.

IV. REQUIRED MATERIALS

As described in Section II.C, the most viable radar for this project is the TI 77GHz AWR2243 radar. The Infineon 60 GHz FMCW radar could also be appropriate for this project, but is underpowered when compared to the above – the main problem is the 10x smaller ADC sampling rate, which significantly eats into the range resolution (or, in other terms, just requires more dwell time, potentially leading to slower sampling).

This project's focus is more built around signal processing algorithms and less so hardware/applications, so either sharing a radar with another team or (less ideally) forgoing a radar and using an online dataset is possible.

An online dataset such as the “Gotcha Volumetric SAR Data Set” (<https://www.sdms.afrl.af.mil/index.php?collection=gotcha>) can be used to accelerate the project.

No other supplies are needed.

REFERENCES

- [1] S. Guo and X. Dong, “Modified Omega-K algorithm for ground-based FMCW SAR imaging,” in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, 2016, pp. 1647–1650. doi: 10.1109/ICSP.2016.7878107.
- [2] T. Xiong, M. Xing, Y. Wang, S. Wang, J. Sheng, and L. Guo, “Minimum-Entropy-Based Autofocus Algorithm for SAR Data Using Chebyshev Approximation and Method of Series Reversion, and Its Implementation in a Data Processor,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1719–1728, 2014, doi: 10.1109/TGRS.2013.2253781.
- [3] A. Bekar, M. Antoniou, and C. J. Baker, “Low-Cost, High-Resolution, Drone-Borne SAR Imaging,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, no. , pp. 1–11, 2022, doi: 10.1109/TGRS.2021.3085235.