Synthetic aperture imaging radar signal simulation of urban environments

Thomas Stenson Moffat
11501

MEng final individual project report

Academic Year 2020-2021
Dr Robert Watson
Wordcount:

Abstract

Contents

List of Figures

List of Symbols

A_e	Effective antenna aperture area
B	Bandwidth of the signal
D	Aperture size in desired dimension
ΔR	Radial resolution
Δr	Difference between r and r_0
ΔR_g	Ground resolution
Δx	Cross-range resolution
G	Total gain
k	Boltzmann's constant
k_0	Angular wave number (i.e. $\frac{2\pi}{r}$)
λ	Wavelength of the signal
n	A number
P_{av}	Average power across all pulses
$\phi(x)$	2-way phase history
P_n	Noise power
P_t	Transmitter power
R	Range
r	Outer radial distance of the beam
r_0	Center distance of the beam
$R_{\max}$	Maximum range
σ^0	Noise equivalent
σ_b	Radar cross-section
θ_{sa}	Synthesised beam width
T_{obs}	Time spent observing the target
T_{sys}	System noise temperature
v_p	Platform velocity
x	Total displacement of the platform

List of Acronyms

CSA	Chirp Scaling
ωKA	Omega-K
LFM	Linear Frequency Modulated
RCMC	Range Cell Migration Correction
RCS	Radar Cross Section
RDA	Range-Doppler
SAR	Synthetic Aperture Radar
STL	STeroLithography

Chapter 1

Introduction

Chapter 2

Engineering Analysis

2.1 Introduction to SAR

Synthetic Aperture Radar (SAR) is a method of radar imaging that allows for a greater dimension of radar aperture than is possible in a single element or an array of elements. This is generally achieved by mounting a radar transceiver to a moving platform and moving it past the target to be imaged however the transceiver can be stationary and the target moved past it. The image uses the relative movement between the platform and the target to build up an image based on the responses received and then time multiplexing all the responses. This movement, and hence the larger aperture created, allows for a finer spatial resolution than can be achieved by more conventional beam scanning radar. These images can be useful for a variety of applications such as remote mapping of the surface of the Earth for glaciology and geology among others. Interferometric SAR can also be used to create elevation maps by taking two measurements from different positions and comparing the differences. Simulating a SAR image is desirable as flight time can be expensive and time consuming to get quality images so defining the flight paths and angles before any imaging is carried out saves both time and money.

There are two main methods of imaging using SAR. These are stripmap and spotlight imaging. Stripmap imaging SAR keeps the radar beam fixed at 90°to the direction of travel of the platform to which it is mounted, giving an even resolution image over the entire imaging area. Spotlight SAR uses beam steering in order to focus on one specific area to increase image resolution in that area, at the expense of image resolution in other areas. This extra time spent focusing on the single area increases the synthetic aperture length for that section only which increases the resolution.

Generally SAR will use a Linear Frequency Modulated (LFM) pulse although other waveforms can be used for different characteristics. LFM waveforms are well suited for radar as they have a large bandwidth while keeping the pulse length relatively short and the waveform envelope magnitude is constant. They also aren't significantly by Doppler shift, at least when it comes to shape. Doppler shift will shift the pulse in time, which is useful for applications covered later [1].

(a) SAR in stripmap operation

(b) SAR in spotlight operation

Figure 2.1: The two main modes of operation of SAR, reproduced from [2]

2.1.1 The SAR radar equation

There are two main approaches to SAR, focused and unfocused SAR. These differ in how they consider shifts in the response phase, as focused SAR only takes into account the Doppler shift across the aperture, while focused SAR takes into account the phase shift in the beam due to the change in range caused by the target passing through the beam. This means that the resolution of the image is determined by the length of the entire synthetic aperture, as opposed to in unfocused SAR where the resolution is a function of the shortest range between the platform and the target as well as the wavelength of the signal. The resolution of focused SAR is wavelength independent. The derivation of this can be found in Appendix A. Also in this appendix is a derivation of the radar equation both generally and then to the more specific version for SAR. The more general form of the radar equation for a single pulse is given as

$$\text{SNR} = \frac{P_t G^2 \lambda^2 \sigma_b}{(4\pi)^3 R^4 k T_{sys} B} = \frac{P_t A_e^2 \sigma_b}{4\pi \lambda^2 k T_{sys} B R^4}$$

which gives the signal-to-noise ratio for a single pulse, where P_t is the transmitted power, G is the gain of the system, A_e is the effective aperture area, λ is the wavelength of the signal, σ_b is the radar cross-section, R is the signal range, k is Boltzmann's constant, T_{sys} is the system noise temperature and B is the bandwidth of the system. This can then be adapted to the SAR-specific radar equation by increasing the number of pulses n to give

$$\text{SNR} = \frac{P_t A_e^2 \sigma_b n}{4\pi \lambda^2 k T_{sys} B R^4}$$

and hence the signal-to-noise ratio of a distributed target is given by

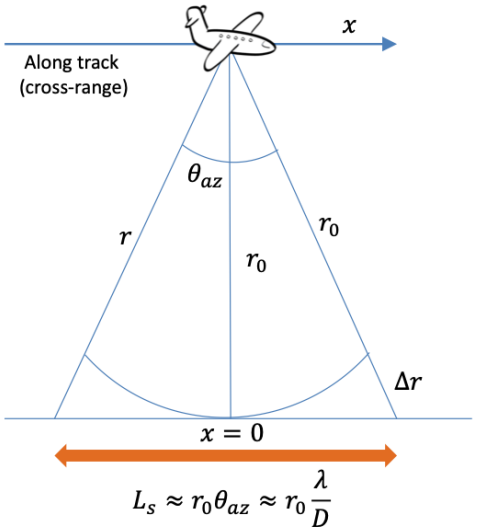
$$\text{SNR} = \frac{P_{av} A_e^2 \sigma^0 \Delta R_g \Delta x t_{obs}}{4\pi \lambda^2 k T_{sys} R^4} \text{ or } \text{SNR} = \frac{P_{av} A_e^2 \sigma^0 \Delta x \lambda^3}{(4\pi)^3 R^3 k T_{sys} 2 v_p}$$

where P_{av} is the average power across all pulses, ΔR_g is the ground range resolution of the system, T_{obs} is the observation time of the target and v_p is the platform velocity. The sensitivity of a SAR system can be expressed as the noise equivalent σ^0, which is the value of σ^0 at the maximum range to give a signal-to-noise ratio of 1 (or 0 dB) [3].

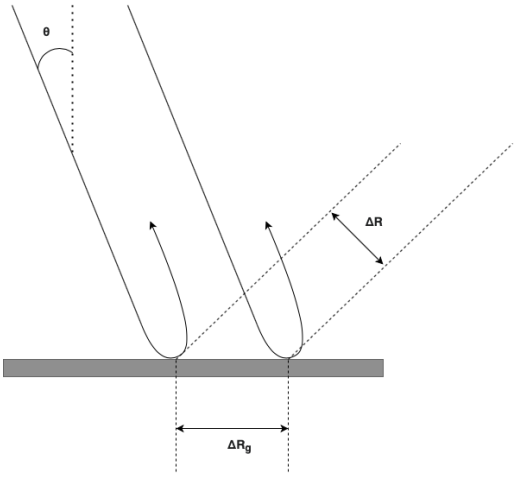
2.1.2 SAR Image Artefacts

There are a few artefacts that can arise as part of SAR imagery (and radar imagery in general). Some of these, such as range ambiguities, aren't relevant in this case due to the fact that all the targets being imaged are static and this is only a real issue in moving objects. Some artefacts will need to be considered though. The first of these is speckle, which is a result of the fact that the energy arriving at the pixel is pretty coherent, so it has a single phase on arrival at said pixel. Generally the pixel will be the sum of multiple scatterers as opposed to one single scatterer, meaning that it is likely to be affected by background reflections and so result in a relatively noisy image. This can be fixed in real world applications by splitting the SAR antenna into multiple sections and take the same image with each section. This then allows for the pixel values to be averaged out, hopefully removing the speckle at the cost of some resolution [4].

Another image artefact that will need to be considered can be caused by strong scatterers that means that the signal is reflected multiple times before returning to the receiver. This can be seen most strongly with a bridge over water as it involves both a hard target in the bridge and a dihedral reflection from the water. This mostly occurs when the target is aligned with the path of the platform and the multiple reflections can cause the bridge



(a) Along-track/Cross-range SAR Resolution, reproduced from [3]



(b) Down-range SAR resolution, adapted from [4]

Figure 2.2: Cross-sections of both dimensions of SAR range

to appear multiple times in the image due to the multiple bounces taken by the signal increase the Doppler shift presented when it is filtered on reception. The same signal is in essence received multiple times and so the algorithm reads it as different displacements for the same target [4].

2.1.3 Other SAR Imaging Issues

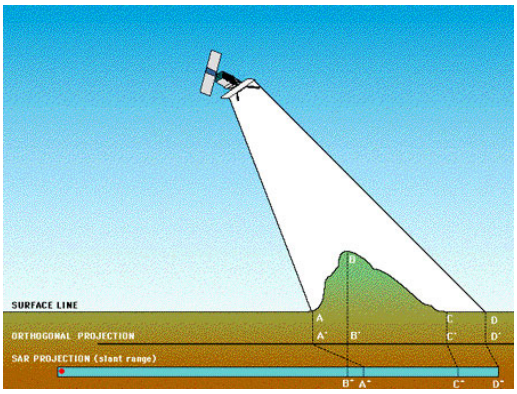
There are three other issues that can occur in a SAR image that aren't really artefacts per se but are issues that normally would need to be taken into account. Two of these issues are caused because SAR measures travel times instead of image angles like in optical systems and the third issue is just thanks to physics. These three issues are layover, foreshortening and shadow and can be affected by the angle of incidence of the radar beam. If this report concerned space-based SAR then there would be very little issue as the smaller range of incidence angles means less image distortion due to them. These issues aren't likely to be relevant in this report however they are included here because they help understand some of the unique characteristics of SAR.

2.1.3.1 Layover

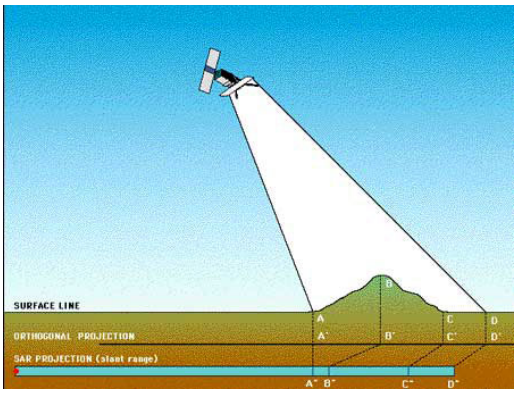
Layover occurs when the top of the object appears in a range before the bottom. This is caused because the top of the target has a similar slant range distance to the bottom of the target, which means that they appear in a reversed order when the image is produced and so the slope is much steeper than it may actually be. This can't be fixed in the image formation stage as the reflections in that area are compressed together and so can't be separated [5]. This can be seen in Figure 2.3a and the bright areas of Figure 2.4a.

2.1.3.2 Foreshortening

Foreshortening is the other issue caused by SAR working differently to optical systems and can be more effectively demonstrated using figure 2.3b. On this, if plotted orthographically then points A, B and C are equidistant however on the SAR image this is not the case as the top of the mountain is closer to the sensor. This can be removed using geocoding if a

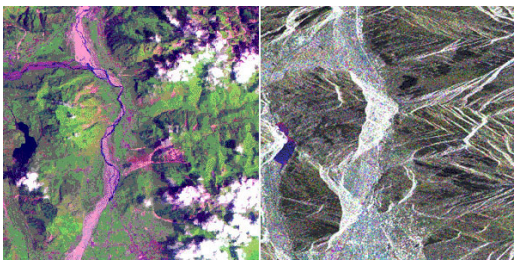


(a) The effects of layover in a SAR image

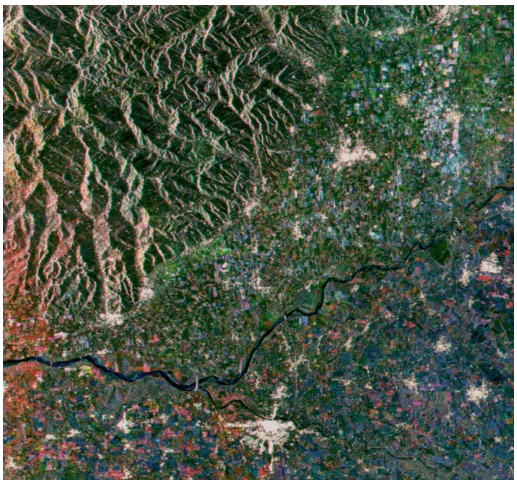


(b) The effects of foreshortening

Figure 2.3: SAR range direction image anomalies [5]

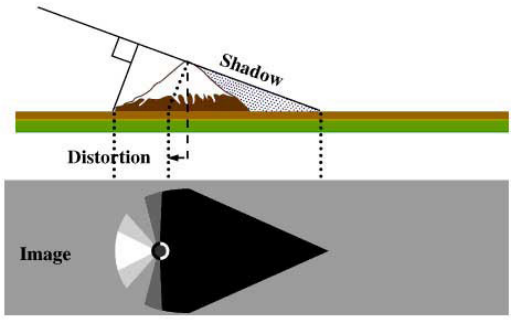


(a) A real image showing layover from near the city of Udine in Italy

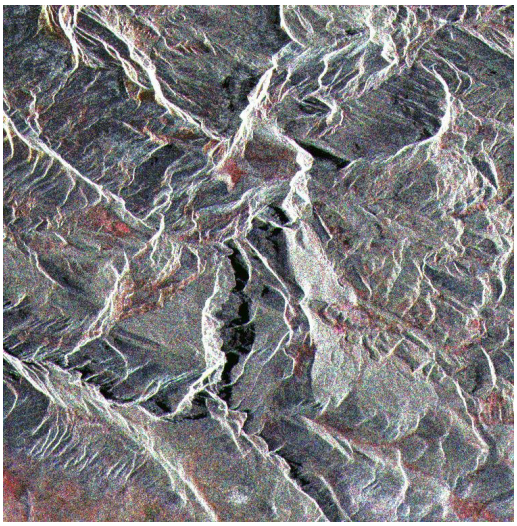


(b) An image showing the effects of foreshortening, area unknown

Figure 2.4: SAR range direction image anomalies [5]



(a) An image demonstrating SAR shadow



(b) A real SAR shadow from the Cote d'Azur area of France, specifically with the Grand Canon du Verdon in the middle of the image

Figure 2.5: SAR Shadows [6]

model of the terrain is available. This is similar to layover but not quite the same, and an image demonstrating it can be seen in figure 2.4b. Notice how the mountains all appear to lean in the same direction, towards the sensor [5].

2.1.3.3 Shadow

Shadows in the SAR image are caused whenever a target in the image slopes away from the radar illumination with a slope greater than the sensor angle of incidence (so it is blocking the illumination). This appears in a SAR image much like a shadow from the sun would in an optical image [6] and images demonstrating this can be seen in figure 2.5.

2.2 Literature Review

2.2.1 A Summary of Methods Previously Explored in Literature

The table that follows summarises the methods used in literature and a brief list of advantages and disadvantages. An in-depth exploration of the papers used can be found in section 2.2.2 and an exploration of each possible method and their suitability for this project can be found in section 3.1.

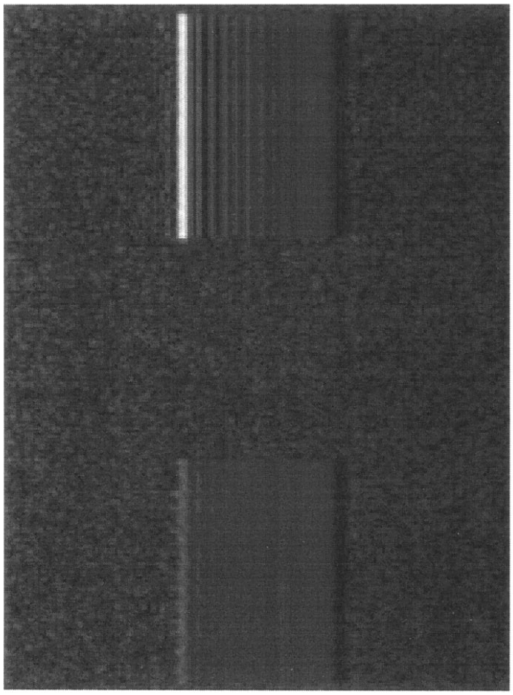
Method	Advantages	Disadvantages
Raw signal	The most accurate method of simulation available as it uses the characteristics of the radar beam in order to simulate the responses of the structures it reflects off.	Most computationally complex method as it has to simulate the effects of every beam every time a pulse is transmitted and received.

Method	Advantages	Disadvantages
Rasterisation	The fastest method of simulating SAR as it uses image transformation techniques in order to give a good idea of how an image will look, this method is capable of real-time updates of the image which means it has applications in SAR data capture planning.	The least accurate method as it is only performing an image transform as opposed to a full simulation of the signal.
Ray tracing	A good balance of speed and accuracy, uses advanced computer graphics techniques to only simulate the parts of the model available to the observer so is less computationally intense than the raw signal but also more accurate than rasterisation.	Requires a good graphics processor as well as a good modelling environment in which to perform it, this is by far the most complex to implement from scratch
Hybrid methods	Has the potential to be a good balance between the raw signal and ray tracing as it uses raw signal for the higher order contributions and ray tracing for the more obvious single reflections	This is the least explored method so very few conclusions about its actual performance can be drawn, with only one paper found that uses it.

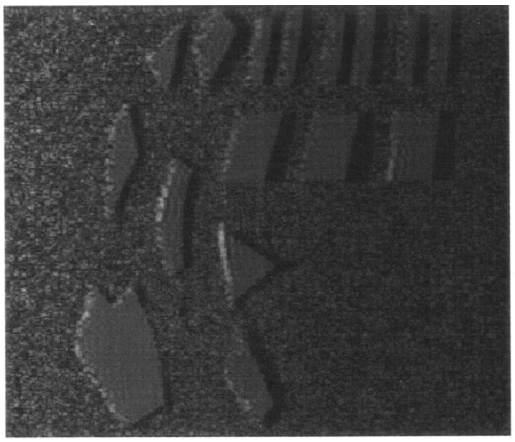
2.2.2 In-Depth Findings

2.2.2.1 Raw SAR Simulation

There are three main methods that have previously been explored for simulating SAR. The first of these, which is the most accurate is simulating the effects of the beam itself, as detailed in Franceschetti *et al.* 2003 [7]. This describes creating the raw data as would be expected in a SAR system in the real world and then transforming it to create a SAR image. The ground work for this was laid in Franceschetti *et al.* 1992 [8], which itself builds on Francescetti and Schirinzi's work on a SAR processor based on two-dimensional FFT codes [9]. This approach has the advantages of being able to take into account backscattering as well as the higher-order signal contributions created by scattering as the signal reflects off of both a wall and the ground. A lot of the preliminary work with regards to the behaviour of electromagnetic backscattering was conducted in [10]. This describes a model that allows for the return from a structure to a microwave sensor to be analysed and so determine its dielectric properties as well as its geometric properties. The optics can be altered to simulate the roughness of the surface the signal is reflected off of last before it is received by the sensor. This can also be expanded to simulate backscattering. This simulator as a whole works very well with individual objects, however the simulation time was found to scale linearly with the number of objects present in the scene. Simulations in [7] were carried out using a Pentium IV processor from Intel Corp, released in 2001 which means that the simulation times achieved are not necessarily reflective of the performance achievable on more modern processors, however at the time it was found that while one object in a 512x512 pixel image required approximately 34 seconds, two objects in the same sized image required 1'02" and 16 objects increased this computation time up to 7'38". This is not ideal if the aim is real-time or even near real-time simulation capabilities as most urban environments that are to be simulated will contain far more than 16 structures. This efficiency does seem to have been improved in Franceschetti *et al.* 2007 [11] as in



(a) Simulated single-look SAR with two buildings



(b) Simulated single-look SAR with 16 buildings

Figure 2.6: Simulated SAR images from Franceschetti *et al.* 2003 [7]

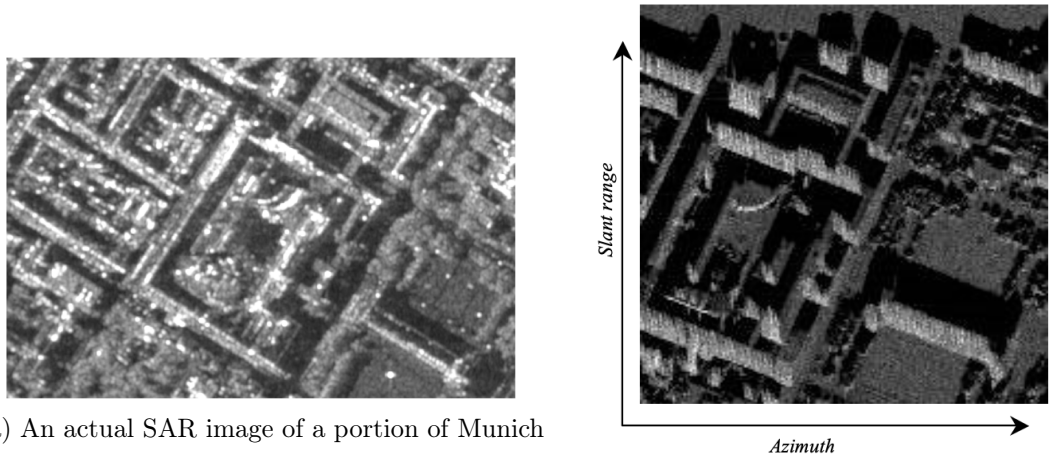
this paper it is used to simulate a SAR image of a $400 \times 600\text{m}^2$ area of the centre of Munich, incorporating the Technische Universität and the Alta Pinakothek, which leads to a reasonably complicated scene.

A similar approach building on the previously mentioned work is presented by Zheng *et al.* [12] which uses the scattering model in order to accurately simulate a SAR scene. This is achieved by making the assumption that the scene is made up of vertical buildings distributed on a rough dielectric terrain. This also discusses the computation of scattering coefficients under different conditions. As with the previous approach (and the following raw SAR approach) the Kirchhoff approach is used in this case.

A similar approach to robust SAR simulation is described in Dellière *et al.* 2007 [13], which uses an electromagnetic approach closely following Maxwell's equations to create a finite-difference time domain method of simulation, which differs from [7] as that instead manipulates the signal in the frequency domain in order to create the raw SAR signal. This system has similar drawbacks to Franceschetti's approach with regards to computation complexity and time required, but is even more computationally complex, due to its ability to handle dispersive materials as well as phase changes. This approach could be of some interest for creating an incredibly robust simulator due to the advances in computing performance between the publication date and today.

2.2.2.2 Graphical SAR Simulation

There are two methods that are less complex computationally for simulating the image achieved by SAR. These both arise from graphics manipulation and more specifically 3 dimensional modelling. These are rasterisation and ray tracing. Rasterisation is the process of taking a 3 dimensional area and creating a 2 dimensional image using some sort of image transformation based on the location of the camera to the object in the 3D area. Ray tracing follows a similar process to light, just in reverse. It sends beams out from the camera in all directions that the camera can see and simulates them reflecting off



(a) An actual SAR image of a portion of Munich

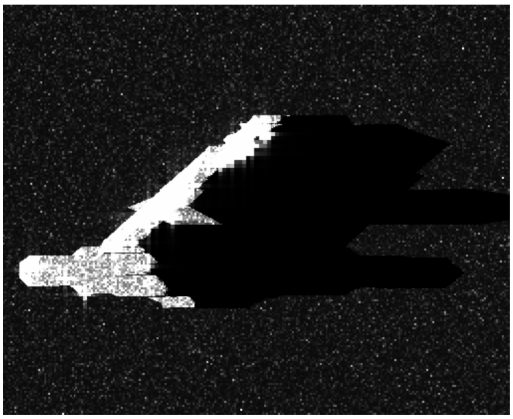
(b) A simulated SAR image of the same area

Figure 2.7: A comparison between an actual SAR image and simulated SAR image of the Technische Universität and Alte Pinakothek from [11]

of objects until they hit a source of illumination. Using this the rendering engine can determine if an object is illuminated or not and adjust appropriately. These two approaches have both been used in the past for simulation of SAR images. Rasterisation was used by Balz 2006 [14] and ray tracing was explored by Auer *et al.* 2008 [15] and Mametsa *et al.* 2002 [16]. This second paper was used in Hammer *et al.* [17] along with Balz 2006 to compare the relative merits of these two approaches. Ultimately the advantages of Balz's approach as detailed in [18] and [14] is that the simulator is real-time so can be used to determine optimal azimuthal directions when recording a SAR image using a physical airborne platform. This form of simulation however doesn't take into account the contributions of higher order reflections so if it is being used for a demonstration tool, the images produced are less representative of an actual SAR image. This also means that corners aren't visible in the image, meaning that this form of simulation can't be used to test feature extraction algorithms. The two ray tracing simulators tested within [17] can simulate these higher-order contributions meaning they can be used to test feature extraction algorithms, however due to the computationally intensive nature of ray tracing these can't be simulated in real time, so this form of simulation is less useful for planning purposes.

All of the forms of simulation presented here so far are only as good as the models used, as generally the modelled buildings are assumed to be made of one material with a fixed dielectric constant, while the modelled ground is made of a different material. In [17], the buildings modelled are created entirely of stone, with a grass ground. In real life however this becomes more complicated as buildings do tend to be made up of multiple materials, and often in urban areas the ground is made up of a material with a similar dielectric constant to the buildings. This also doesn't include any forms of dispersive materials such as metal. It is unclear whether any of the simulators covered in [17] can simulate these sorts of materials accurately, which is something that should be taken to account in the future.

Yet another method similar to the image processing approaches is presented by Lu *et al.* [19] and uses methods most commonly used by video games. In this case the raw mathematic SAR simulation is not carried out, similar to [20] and [15] and uses an orthogonal projection to cast the 3D model to a 2D image based on the slant range of the camera. This again has the potential of improving the performance of image generation in

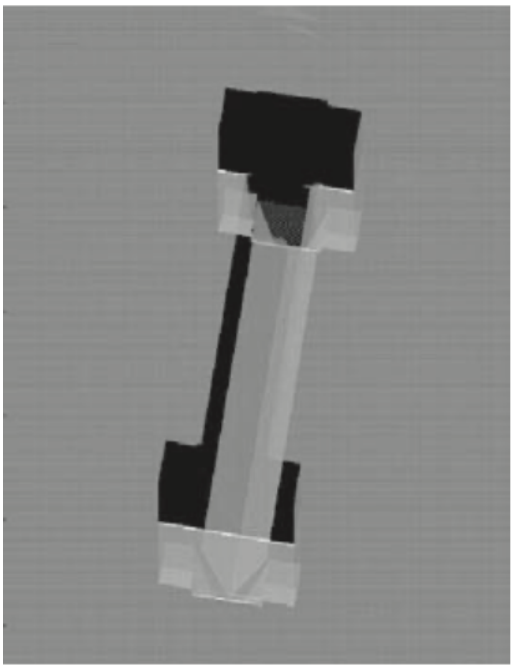


(a) 0.3m resolution simulation of the Stiftskirche in Stuttgart

(b) 5m resolution simulation of Stuttgart

Figure 2.8: Results from Balz 2006 [18] using rasterisation techniques

(a) A reference spotlight image of the Alta Pinakothek



(b) A simulated SAR image using ray tracing of the Alta Pinakothek

Figure 2.9: Simulated SAR using ray tracing from Auer *et al.* 2008 [15]

incredibly complicated scenes such as city centres.

2.2.2.3 Hybrid Method and Other Papers of Note

Chen *et al.* 2011 [21] proposes a hybrid method of simulating these images using analytical models. In this approach, the contributions provided by backscattering are shown using an electromagnetic analytical model ultimately based on that described by [7], and the object position vectors are given by a geometric model using ray tracing. In this paper the analysis is mostly focused on cylindrical or cylinder-like buildings, with some analysis devoted to flat-roof buildings as well. This approach has a potential to be more accurate than the ray tracing models proposed previously due to the electromagnetic models involved, but also be less computationally complex than the more canonical SAR image simulators.

Xu and Jin 2006 [22] is a paper that is not immediately relevant to this situation due to it modelling natural environments for SAR simulation however this is potentially interesting and relevant due to the variety of materials present in urban environments to be modelled. This is an algorithm that can deal with randomly and heterogeneously distributed objects within an image with varying dielectric constants and can deal with them in a reasonable length of time. This is something that is worth exploring if there is time in the future as it would add robustness to the system as a whole.

Chapter 3

Methods

3.1 Exploration of Potential Methods and Their Suitability

The four possible methods that can be used in this project are, as mentioned previously, raw signal simulation, rasterisation, ray tracing and a hybrid method. In this section I will cover their suitability as they relate to this project, as well as explore the methods of transforming SAR raw signals into an image.

3.1.1 SAR Simulation Methods

3.1.1.1 Raw SAR Simulation

This is the most robust form of simulation and the most accurate as the individual signals are simulated as well as their reflections. This does have some drawbacks as developing the system in the first place is likely to be quite complex. This system requires the implementation of a transmission and reception element (likely to be two separate functions for complexity reasons), a free-space element, an element to act as the target would in real life to accurately simulate the return signal, some sort of coordinate system to position both the target and the platform in free space and finally the linear FM waveform. These are all sections that, while the conceptual understanding of their operation is relatively easy to grasp given prior knowledge of how wireless communications and more specifically radar work, are not altogether simple in their implementation. Fortunately this would not be implemented blind thanks to Franceschetti *et al.* 2003 [7] and all the foundational work that led to it however this work has been refined over the course of more than a decade, not several weeks, which is potentially the biggest limitation on how in-depth this implementation can go and the biggest roadblock against this particular method of simulation.

3.1.1.2 Rasterisation

As previously mentioned, rasterisation is one approach to SAR simulation that does not require lengthy implementation processes as rasterisation is a standard process in image manipulation. In this case what would be required is devising some sort of transform from the 3-dimensional image simulated and the SAR output. This will not be true SAR as it is only the result of taking one image from one place and therefore will not truly be an accurate representation. As was discussed in Balz 2006 [18] and all related work this is a desirable outcome in the event that this SAR simulation is to be used as a real-time simulation in order to plan actual SAR imaging flights. In this project this isn't necessarily the case as my intention is to create a final product that is usable as a teaching aide, which would involve some sort of progression as the SAR image is taken. This isn't a particularly

practical approach to be used with this in mind as it can be done instantly (or as near as to a human) and the image isn't built up over multiple measurements.

3.1.1.3 Ray-tracing

Ray-tracing is, in this case, somewhat of a hybrid method. As it doesn't simulate the free space and electromagnetic effects of the radar it isn't truly SAR however it works in a similar just inverted way. This is because the light beams (here analogous to the higher frequency EM waves) are projected from the observer (the receiver) and reflect off objects until they reach a source of illumination (the transmitter). This could be useful for teaching purposes as the method by which it works could be the same and then the image is built using similar image formation methods to those used by true SAR. The issue with this type of simulation is a relative lack of experience in it and some issues with the complexity of implementation. This is partially an issue with all three forms of simulation which will be covered in the following section, but also ray-tracing is known to be a very computationally intense form of visual rendering and requires a strong knowledge with three-dimensional rendering techniques to even begin to implement a version of this.

3.1.1.4 True Hybrid Methods

This would probably be the most complex path to take as it would require choosing two of the above methods and combining them in an as yet mostly unexplored way. This would run most heavily into the issues laid out in section 3.1.1.5 as it would require heavily editing the 3D rendering engine chosen to be able to perform the necessary simulations which would reduce computing resources required in the long run but would possibly require more work than is possible during the ten weeks allotted to this project.

3.1.1.5 3D Rendering Engines

There is a problem common to all three of these potential methods, which is that no matter what they will require some form of 3D rendering engine, either to simulate the effect of the radar waves or ray-tracing, or to provide a 3D image base for the rasterisation transformation. Unsurprisingly none of the rendering engines currently commonly available are perfectly suited for this and many of the ones investigated were unusable for this case. Ideally some time would be allowed for creating a custom 3D modelling engine that would properly handle EM waves or equivalent. The closest to being feasible for this application appeared to be Blender [23], an incredibly powerful modelling engine released under a GNU Public License by the Blender Foundation that has a Python scripting engine, however due to a lack of familiarity with this software it was decided that gaining a strong enough familiarity with the package would take too long to be practical in this project. The final approach chosen is discussed in Section 3.1.3

3.1.2 Image Forming Methods

Raw SAR simulation requires an image formation method. This takes the overlapping received signals and time-multiplexes them to display an image that can be interpreted by a person, especially with multiple targets at the same range. There are multiple forms available and a few of them will be discussed here. These will all be the forms used for airborne (including space-based) platforms as there are some corrections required for ground-based systems that are irrelevant here although are covered in Guo and Dong (2016) [24], as well as in a practical sense on the blog of Henrik Forstén for both an Omega-K [25] and Backprojection [26] approach (both covered here, among others). All the algorithms will operate in similar ways however they all have strengths and weaknesses.

3.1.2.1 Omega-K (ωKA)

Omega-K is otherwise known as the range migration algorithm. It works by first Fourier transforming the received SAR signal into the frequency domain and then performing bulk compression, which is where the signal is multiplied by a reference function computed for a known range. A target is correctly focused at this reference range but targets not at this range are only partially focused. Following this, the signal undergoes Stolt interpolation, which re-maps the range frequency axis in order to bring targets away from the reference range into focus. It achieves this by transforming the range frequency so the phase is linear. This effectively removes all phase terms higher than the linear term, although these aren't ignored. Finally an inverse Fourier transform is performed on the signal to put it back into the image domain. For a more in depth explanation of how the Stolt mapping works, see Cumming *et al.* 2003 [27, section 3]. This approach does assume that the trajectory is completely straight so if this isn't achieved then the displayed image will be incorrect [28]. For this application this is not an issue of concern as the simulated airborne platform will always be travelling in a straight line but is worth keeping in mind for more general applications.

3.1.2.2 Backprojection

Backprojection uses a matched filter approach to get the difference between the expected return from the radar and the actual return from the radar and was originally developed for use in computer-aided tomography for medical reasons [29]. This is generally performed in the time domain as opposed to the frequency domain due to the lack of assumptions that have to be made [28]. The way this works is the position of the platform is known in 3 dimensional space and so the expected return can be calculated based on the time delay from the reflected signal. The matched filter that this creates can then remove this from the received signal and the difference is due to an object that reflected the signal prematurely. There are some other sources of error in this case such as multiple bounces on the image or impreciseness in calculating the imaging platform position [1]. This algorithm is not ideal as it is very computationally intensive and requires that the imaging geometry needs to be precisely known. This positioning can normally be estimated using an inertial navigation system so is not particularly difficult to obtain however it does need to be precise as this can make received images appear shifted from where they are actually positioned. As each pixel can be handled independently of the other pixels it is ideally suited for parallel computing so the computation time can be reduced significantly [28].

3.1.2.3 Range-Doppler (RDA)

This algorithm is similar in process to ωKA. It is implemented in the frequency domain but operates only in one dimension at a time, which improves the simplicity of the algorithm. The first step is range migration, which is where the raw data is Fourier transformed in the range dimension and multiplied with the range reference signal and transforms back to the image domain. This algorithm then performs a Fourier transform in the azimuth only, transforming to the range Doppler domain before performing a Range Cell Migration Correction (RCMC). This is to correct for the slant range to the target changing as the platform moves through space, meaning that the Doppler frequency of the target is different at different times and so appears at different ranges. The RCMC corrects for this by shifting each cell a set number of metres depending on its distance from the reference range [30]. After this a matched filter for just the azimuth is used to perform azimuth compression and the whole image is returned to the image domain for visualisation [31]. This algorithm was created in the 1970s and so is more basic than the Omega-K algorithm. In this case the RDA in an accurate form is somewhat equivalent to an approximate form of ωKA,

however RDA is often used in a less accurate form where the range compression is not varied according to either range or azimuth frequencies, and the accurate form of ωKA is more accurate than the approximate form [27], so is a better choice for this application.

3.1.2.4 Chirp Scaling (CSA)

Chirp scaling is quite similar to RDA, and the difference is improving the RCMC by removing the interpolation required. It does this by applying a phase multiply in order to equalise the range migrations of all the targets. This Chirp scale factor is generally 1 for a satellite based SAR system and higher for aircraft based [31]. This algorithm has an accuracy somewhere between that of the accurate and approximate ωKA and approximate ωKA is roughly equivalent to CSA if there was no chirp scaling performed [27].

3.1.3 Discussion of Approaches Chosen and Reasoning

After carefully considering the approaches discussed above, the decision has been made to implement the SAR simulation using raw methods, specifically using the Simulink Phased Array Toolbox, with image formation performed using MATLAB. This is because the issues with implementing all the sections mentioned previously are removed and they are created in a far more robust way than would be possible to implement within the time available for this project. Simulink has been chosen for this because it's more tactile than the equivalent MATLAB code would be so is better for the desired outcome in this case which is to create a tool that can be used as a teaching aid while learning synthetic aperture radar. Simulink also gives access to the Simulink 3D Animation Toolbox, which allows for linking models to a 3D visualisation, which is similarly useful in a teaching environment. Simulink is not suited to image formation algorithms as they require all the data to be present at once, whereas Simulink works best when the data can be fed in over the length of the simulation. Therefore these will be implemented in MATLAB. In this case, both ωKA and backprojection will be implemented as the differences between the two will be interesting to observe, especially considering the data input will be identical. These have been chosen as they are more accurate than either CSA or RDA and so will give a better representation. Specifically ωKA will be implemented using the accurate version described in Cumming *et al.* 2003 [27] and backprojection will be implemented based on the efficient approach outlined in Yegulalp 1999 [32] as this should be simpler to implement than more standard backprojection algorithms, especially as the steps are very kindly laid out within the paper. The way this version of backprojection works is by dividing the full aperture into subapertures and generating an image in each of the subapertures. These can then all be recombined by coherently summing all the images into the final image. All of the subapertures are quite coarse in the cross-range direction which reduces the number of operations required and then they can be upscaled when they are combined back into the full image.

3.2 Desired Results and Outcomes

The major desired outcome of this project is to have a Simulink model that can simulate a SAR system with one target, in such a way that it can be easily examined for teaching purposes. This model must output data so that a MATLAB script can read it in and perform at least one and more ideally two methods of image formation. This Simulink model must also feed into a Simulink 3D World in such a way that it is obvious what is happening in the model and provide a visual aid to the model when it is running. Extended from this it would be ideal if the dimensions of the target could be read in from the Simulink 3D World and used to update the Radar Cross Section (RCS) of the target

within the Simulink model in a dynamic fashion. As another extension to this, a cone to visualise the antenna beam within the Simulink 3D World would be nice to have however is not completely necessary. Finally, the ability within the model to switch between stripmap and spotlight modes and have the cone within the 3D World update to match would be nice however again not strictly necessary.

3.3 Milestones

These milestones have been revised from the original milestones as set out in the interim report [33]. This is due to an increase in understanding of the requirements of the project and an updated schedule.

Milestone	Description
1 - Basic Model	The first milestone is to have a SAR system model that outputs data in a usable format and can be visualised using a MATLAB plot function. The target in this case can be just a spot target with a high average radar cross section as the realism is not a worry in this case.
2 - Basic Image Formation	In this milestone the aim is to convert the data output by the model in Milestone 1 to a visible image using some form of image formation algorithm. This can be either ωKA or backprojection, although it will most likely be ωKA due to the comparative simplicity in implementation when compared to backprojection. The other will be implemented following Milestone 3 as it is less important to have two different image formation algorithms than this step.
3 - Basic Reference Model	Milestone 3 is to implement a 3 dimensional representation of the process that is occurring in the model. At this stage the only real requirement is to have a representation of the airborne platform and a representation of the target in the correct places for a given point in the simulation. This will also include the platform moving at an accurate rate past the target.
4 - Completed Image Formation	The aim of this milestone is to finish the image formation algorithms selected for use in this model by implementing whichever algorithm is not created in milestone 2.
5 - Simulink Model Improvements	The aim of this milestone is to implement the other major form of SAR to that created in milestone 1, whether that is stripmap or spotlight and to include a method of being able to display the results of both forms of SAR at one time or another. The form that this will take is yet to be decided.
6 - 3D Model Improvements	Improve the amount of information imparted by the 3D model for an educational environment by representing the antenna beam or similar
7 - Immersiveness	Use a calculated radar cross section of the target to improve the image while also displaying this target within the 3D environment

Chapter 4

Deliverables

4.1 The Final Product

4.1.1 Deliverables Provided

The system created is not the most user-friendly piece of code ever created however it does the job it was created to do, to simulate the SAR imaging of a building

4.1.2 Milestones Achieved

4.2 Significant Results

4.3 Uncertainty

4.4 Other Major Issues Encountered

Chapter 5

Conclusions

5.1 Final Remarks and Outcome

5.1.1 Project Review

5.2 Further Work

There is a significant amount of work that can be done on this model to improve it.

5.2.1 Arbitrary Radar Cross Sections

The most interesting improvement that could be made to the system would be the creation of an RCS solver for the target model that can take into account the materials used when 3D modelling the target structure. This would allow for a much more realistic target output and would allow for arbitrary targets to be used. To begin with this could use a single material similar to the built-in MATLAB version but then could be expanded to take into account multiple materials and how they affect radar cross sections. This would require the use of a more advanced 3D model file format than the STeroLithography (STL) file used by the MATLAB solver as STL files only relate the vertices of the modelled object and not the textures or materials applied to them in the modelling process. A file format like the 3DS format created by Autodesk would appear to work well for this purpose as this format is able to preserve textures. This could then use data acquired (either through modelling or experimental acquisition) on the electromagnetic reflectivity of a variety of materials to map a more exact RCS than has previously been possible. This is one possible alternative as using a pre-calculated RCS of the building may be limiting. Depending on the efficiency, the solver could be able to re-calculate the RCS in a short enough time that the demonstration model could still be run in a reasonable (within five minutes or so for a four second simulation) amount of time in order to be usable in a lecture or self-learning environment.

5.2.2 A More Robust Approach to Simulation

Really the biggest potential improvements to this model come from re-implementing every component from scratch. This would likely require a lot of work however the ability to be able to run an essentially bespoke SAR simulator would, in my opinion, probably be worth the effort. This would mean that the limitations placed on this project by using the pre-existing MATLAB blocks would be removed as the blocks could be reconfigured to run however they were needed to. This would require re-examining of every component and reimagining its function in a way that is hopefully easier to follow. The Unreal Engine is a video game engine released by Epic Games that is free to download and has been used

by hundreds of video games, including Fortnite Battle Royale, since it was released in its first version in 1998. MATLAB already has a link into Unreal Engine 4 in the Vehicle Dynamics Toolbox and it should be possible to reverse engineer this link for use in other applications or otherwise Unreal Engine uses C++ as its programming language.

What this means is that Unreal Engine would provide a useful platform to build a custom system that could work in a way much closer to that described in Franceschetti *et al.* 2003 [7], with radar pulses transmitted from an aeroplane across free space, to a target with the radar signature calculated in near-real time and then returned to the airborne platform. In this, every aspect of the system would be clear and known to the person implementing it and can then be related to the end user. It is still unclear whether MATLAB calculates higher orders of reflection than the first order and due to the failure of the RCS solver I was unable to test this. Using a system that was created bespoke these unknowns can be controlled for and explicitly implemented or excluded. The other advantage of this system would be that multiple approaches discussed at the beginning of the project can be directly compared against each other which has been done before in Hammer *et al.* 2008 [17] however to my knowledge it has never been done when all the source data is in the same rendering engine and also hasn't included raw SAR simulation before. This could be a very interesting form of comparison to see just how accurate the three forms of simulation are when compared to a reference SAR image and an accurately modelled 3D environment of the area the reference SAR image was captured over.

Using the Unreal Engine or another video game engine would also mean that the quality of visual aids would be improved. Both on a graphical fidelity standpoint but also being able to see more of what the aircraft is doing at any one time. A pre-rendered demonstration could be produced in order to show individual pulses and then the section that reflects what's actually going on within the simulation model could be improved massively as well just due to more powerful tools. The downside to this is there would be more 3D modelling involved with the creation of this set up however there are many shape libraries already existing that are compatible with video game engines, including a first party store provided by Epic Games for the Unreal Engine. This would reduce the requirement to create new 3D models from scratch as pre-existing assets can always be altered for those purposes.

These improvements can also take the form of an improved background and terrain as well. The current system can only really every have a flat background as that's all the system can take into account. Yes it has coordinates in three dimensions however this doesn't make a lot of difference in the overall simulation, whereas with structures at different elevations this system can also model layover, foreshortening and shadow. This can further aid the comprehension of what, without visual aids, can be somewhat difficult concepts to grasp. Along with this, being able to simulate the effects of reflections off of different materials and multiple reflections can cause natural speckle in the resulting image as opposed to having to artificially induce speckle.

5.2.3 Developing Hybrid Methods

Acknowledgements

Bibliography

[1] M. I. Duersch, "Backprojection for Synthetic Aperture Radar," Brigham Young University, Provo, 194 pp.

[2] C. Wolff. (). "Synthetic Aperture Radar Modes," Radartutorial, [Online]. Available: https://www.radartutorial.eu/20.airborne/ab08.en.html.

[3] R. Watson, "EE40136: Radar Systems and Remote Sensing," Lecture Notes, Lecture Notes, University of Bath, Bath.

[4] J. Richards, *Remote Sensing with Imaging Radar*, ser. Signals and Communication Technology. Springer, ISBN: 978-3-642-02019-3.

[5] *Foreshortening and Layover*, in *ERS Radar Course 3*, ESA Operational EO Missions, European Space Agency. [Online]. Available: https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/ers/instruments/sar/applications/radar-courses/content-3/-/asset_publisher/mQ9R7ZVkKg5P/content/radar-course-3-slant-range-ground-range.

[6] *Shadow*, in *ERS Radar Course 3*, ESA Operational EO Missions, European Space Agency. [Online]. Available: https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/ers/instruments/sar/applications/radar-courses/content-3/-/asset_publisher/mQ9R7ZVkKg5P/content/radar-course-3-shadow.

[7] G. Franceschetti, A. Iodice, D. Riccio and G. Ruello, "SAR raw signal simulation for urban structures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, p. 10,

[8] G. Franceschetti, M. Migliaccio, D. Riccio and G. Schirinzi, "SARAS: A synthetic aperture radar (SAR) raw signal simulator," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 1, pp. 110–123, ISSN: 1558-0644. DOI: 10.1109/36.124221.

[9] G. Franceschetti and G. Schirinzi, "A SAR processor based on two-dimensional FFT codes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 2, pp. 356–366, ISSN: 1557-9603. DOI: 10.1109/7.53462.

[10] G. Franceschetti, A. Iodice and D. Riccio, "A canonical problem in electromagnetic backscattering from buildings," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 8, pp. 1787–1801, ISSN: 1558-0644. DOI: 10.1109/TGRS.2002.802459.

[11] G. Franceschetti, R. Guida, A. Iodice, D. Riccio, G. Ruello and U. Stilla, "Simulation Tools for Interpretation of High Resolution SAR Images of Urban Areas," in *2007 Urban Remote Sensing Joint Event*, pp. 1–5. DOI: 10.1109/URS.2007.371841.

[12] Y. Zheng, B. Zhou and Z. Zong, "Simulation method of SAR raw echo for urban scene," in *2008 International Conference on Radar*, pp. 503–507. DOI: 10.1109/RADAR.2008.4653976.

[13] J. Delliere, H. Maitre and A. Maruani, "SAR measurement simulation on urban structures using a FDTD technique," in *2007 Urban Remote Sensing Joint Event*, pp. 1–8. DOI: 10.1109/URS.2007.371837.

[14] T. Balz, "Real-time SAR simulation of complex scenes using programmable Graphics Processing Units."

[15] S. Auer, S. Hinz and R. Bamler, "Ray Tracing for Simulating Reflection Phenomena in SAR Images," in *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, vol. 5, pp. V - 518-V –521. DOI: 10.1109/IGARSS.2008.4780143.

[16] H.-J. Mametsa, F. Rouas, A. Berges and J. Latger, "Imaging radar simulation in realistic environment using shooting and bouncing rays technique," in *SAR Image Analysis, Modeling, and Techniques IV*, vol. 4543, International Society for Optics and Photonics, pp. 34–40. DOI: 10.1117/12.453975.

[17] H. Hammer, T. Balz, E. Cadario, U. Soergel, U. Thoennessen and U. Stilla, "Comparison of SAR simulation concepts for the analysis of high-resolution SAR data," in *7th European Conference on Synthetic Aperture Radar*, pp. 1–4.

[18] T. Balz and N. Haala, "Improved Real-Time SAR Simulation in Urban Areas," in *2006 IEEE International Symposium on Geoscience and Remote Sensing*, pp. 3631–3634. DOI: 10.1109/IGARSS.2006.930.

[19] Y. Lu, K. Wang, X. Liu and W. Yu, "A GPU based real-time SAR simulation for complex scenes," in *2009 International Radar Conference "Surveillance for a Safer World" (RADAR 2009)*, pp. 1–4.

[20] T. Balz and U. Stilla, "Hybrid GPU-Based Single- and Double-Bounce SAR Simulation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 10, pp. 3519–3529, ISSN: 1558-0644. DOI: 10.1109/TGRS.2009.2022326.

[21] H. Chen, Y. Zhang, K. Tang, W. Xiong and C. Ding, "Radar imaging simulation for typical urban structures based on analytical models," in *Proceedings of 2011 IEEE CIE International Conference on Radar*, vol. 2, pp. 1362–1365. DOI: 10.1109/CIE-Radar.2011.6159811.

[22] F. Xu and Y. Jin, "Imaging Simulation of Polarimetric SAR for a Comprehensive Terrain Scene Using the Mapping and Projection Algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3219–3234, ISSN: 1558-0644. DOI: 10.1109/TGRS.2006.879544.

[23] B. Foundation. (). "Blender.org - Home of the Blender project - Free and Open 3D Creation Software," blender.org, [Online]. Available: https://www.blender.org/.

[24] 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)*, pp. 1647–1650. DOI: 10.1109/ICSP.2016.7878107.

[25] H. Forstén. (). "Synthetic-aperture radar imaging," Henrik's Blog, [Online]. Available: https://hforsten.com/synthetic-aperture-radar-imaging.html.

[26] ——, (). "Backprojection Backpropagation," Henrik's Blog, [Online]. Available: https://hforsten.com/backprojection-backpropagation.html.

[27] I. Cumming, Y. Neo and F. Wong, "Interpretations of the omega-K algorithm and comparisons with other algorithms," in *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*, vol. 3, Toulouse, France: IEEE, pp. 1455–1458, ISBN: 978-0-7803-7929-9. DOI: 10.1109/IGARSS.2003.1294142.

[28] M. Albuquerque, P. Prats and R. Scheiber, "Applications of Time-Domain Back-Projection SAR Processing in the Airborne Case,"

[29] Y. Na, Y. Lu and H. Sun, "A Comparison of Back-Projection and Range Migration Algorithms for Ultra-Wideband SAR Imaging," in *Fourth IEEE Workshop on Sensor Array and Multichannel Processing, 2006.*, pp. 320–324. DOI: 10.1109/SAM.2006.1706146.

[30] A. Parashar, "A Study on Range Cell Migration Correction in SAR Imagery and MATLAB Implementation of Algorithms," National Institute of Technology, Rourkela, India.

[31] N. Dastgir, "Processing SAR data using Range Doppler and Chirp Scaling Algorithms," Royal Institute of Technology, Stockholm, Sweden.

[32] A. F. Yegulalp, "Fast backprojection algorithm for synthetic aperture radar," in *Proceedings of the 1999 IEEE Radar Conference. Radar into the Next Millennium (Cat. No.99CH36249)*, pp. 60–65. DOI: 10.1109/NRC.1999.767270.

[33] T. Moffat, "Synthetic aperture imaging radar signal simulation of urban environments interim report," Report, University of Bath, Bath.

Appendix A

Derivations

A.1 Focused SAR Resolution

A.1.1 Along track resolution

$$r = r_0 + \Delta r \tag{A.1}$$

$$\Delta r = (r_0^2 + x^2)^{\frac{1}{2}} - r_0 \tag{A.2}$$

$$\Delta r \approx \frac{x^2}{2r_0} \tag{A.3}$$

$$\phi(x) = -2k_0\Delta r = -\frac{2\pi x^2}{\lambda r_0} \tag{A.4}$$

$$\theta_{sa} \approx \frac{\lambda}{2L_s} \tag{A.5}$$

$$\Delta x = \theta_{sa} r_0 \tag{A.6}$$

$$\Delta x = \frac{\lambda}{2L_s} r_0 = \frac{\lambda}{2r_0 \frac{\lambda}{D}} r_0 \tag{A.7}$$

$$\Rightarrow \Delta x = \frac{D}{2} \tag{A.8}$$

In this case, r is the outer radial distance of the beam, r_0 is the center distance of the beam, Δr is the difference between these two distances, x is the distance travelled by the platform over the time measured, k_0 is the angular wave number of the signal, $\phi(x)$ is the two-way phase history of the signal, λ is the wavelength of the signal, θ_{sa} is the synthesised beam width obtained after azimuth processing, Δx is the resolution of the system and D is the aperture size in the chosen dimension (i.e. here it's the azimuthal dimension). Derivation from [4] and [3].

A.1.2 Down-range resolution

With the assumption made that there is some form of pulse compression on the signal we have a radial resolution of

$$\Delta R = \frac{c}{2B}$$

and by geometry we get a ground resolution of

$$\Delta R_g = \frac{c}{2B\sin(\theta)}$$

so to avoid any range ambiguities,

$$\frac{c}{2R_{max}} > \text{PRF}$$

where $R_{\max}$ is the maximum range achievable by the radar, ΔR_g is the ground range resolution and ΔR is the radial resolution of the signal.

A.2 Derivation of the Radar Equation

Again derivations from [4] and [3].

$$\text{Isotropic Antenna Power Density} = \frac{P_t}{4\pi R^2} \tag{A.9}$$

$$\text{Directive Antenna Power Density} = \frac{P_t G_t}{4\pi R^2} \tag{A.10}$$

$$\text{Power Intercepted by Target} = \frac{P_t G_t}{4\pi R^2}\sigma_b \tag{A.11}$$

$$\text{Power density at receiver} = \frac{P_t G_T}{4\pi R^2}\sigma_b \times \frac{1}{4\pi R^2} \tag{A.12}$$

$$P_r = \frac{P_t G_t}{4\pi R^2}\sigma_b \times \frac{1}{4\pi R^2} \times A_e \tag{A.13}$$

$$G_r = \frac{4\pi}{\lambda^2} A_e \tag{A.14}$$

$$G_t = G_r = G \tag{A.15}$$

$$\Rightarrow P_r = \frac{P_t G^2 \lambda^2}{(4\pi)^3 R^4}\sigma_b \tag{A.16}$$

Noise power P_n is given by $P_n = kT_{sys}B$ and so the equation for SNR is given as

$$\frac{P_r}{P_n} = \frac{P_t G^2 \lambda^2 \sigma_b}{(4\pi)^3 R^4 k T_{sys} B}$$

and so the radar equation for a distributed target is given by

$$\sigma_b = \sigma^0 \Delta R_g \Delta x \tag{A.17}$$

$$\Delta x = \frac{R\lambda}{2L_s} = \frac{D}{2} \tag{A.18}$$

$$\Delta R_g = \frac{c}{2B\sin(90^\circ - \alpha)} \tag{A.19}$$

$$t_{obs} = nT_s = \frac{L_s}{v_p} \tag{A.20}$$

$$n = \frac{L_s}{T_s v_p} \tag{A.21}$$

$$P_t = \frac{P_{av} T_s}{\tau} \tag{A.22}$$

$$\tau B \approx 1 \tag{A.23}$$

$$\text{SNR} = \frac{P_{av} A_e^2 \sigma^0 \Delta R_g \Delta x t_{obs}}{4\pi\lambda^2 k T_{sys} R^4} \text{or} \text{SNR} = \frac{P_{av} A_e^2 \sigma^0 \Delta x \lambda^3}{(4\pi)^3 R^3 k T_{sys} 2 v_p} \tag{A.24}$$

Appendix B

Code Listings

B.1 trigger_script.m

```
clear all

load('sar_consts.mat');

model = 'sar_model';
load_system(model);
simMode = get_param(model, 'SimulationMode');
flightTime = str2double(get_param(model, 'StopTime'));

reference_range = 800;
cross_range_resolution = 1;

sim_out = sim(model, 'SimulationMode', simMode);

figure(1)
imagesc(real(reshape(sim_out.spotlight_data, [], 4001))); title('SAR Raw
↪  Spotlight Mode Data')
xlabel('Cross-range')
ylabel('Down-range')

figure(2)
imagesc(real(reshape(sim_out.stripmap_data, [], 4001))); title('SAR Raw
↪  Stripmap Mode Data')
xlabel('Cross-range')
ylabel('Down-range')

matched_filter_coeffs = sim_out.lin_fm_coeffs(1:360);

pulseCompression = phased.RangeResponse('RangeMethod', 'Matched filter',
↪  'PropagationSpeed', c, 'SampleRate', fs);
[spotlight_data, spotlight_range_grid] =
↪  pulseCompression(sim_out.spotlight_data, matched_filter_coeffs);
[stripmap_data, stripmap_range_grid] =
↪  pulseCompression(sim_out.stripmap_data, matched_filter_coeffs);

spotlight_reshaped_data = reshape(spotlight_data, 2002, []);
```

```
reshaped_spotlight_range_grid = reshape(spotlight_range_grid, 2002, []);
stripmap_reshaped_data = reshape(spotlight_data, 2002, []);
reshaped_stripmap_range_grid = reshape(spotlight_range_grid, 2002, []);

figure(3)
imagesc(real(spotlight_reshaped_data)); title('SAR Range Compressed Data -
↪  Spotlight')
xlabel('Cross-range')
ylabel('Down-range')

figure(4)
imagesc(real(stripmap_reshaped_data)); title('SAR Range Compressed Data -
↪  Stripmap')
xlabel('Cross-range')
ylabel('Down-range')

omega_k_image_spotlight = omegak(spotlight_reshaped_data, fs, maxRange, fc,
↪  flightTime, speed, reference_range, prf);
omega_k_image_stripmap = omegak(stripmap_reshaped_data, fs, maxRange, fc,
↪  flightTime, speed, reference_range, prf);

figure(5)
imagesc((abs(omega_k_image_spotlight.')));
title('Spotlight SAR Data focused using Omega-K algorithm')
xlabel('Cross-Range Samples')
ylabel('Range Samples')

figure(6)
imagesc((abs(omega_k_image_stripmap.')));
title('Stripmap SAR Data focused using Omega-K algorithm')
xlabel('Cross-Range Samples')
ylabel('Range Samples')

backproject_image_spotlight = backprojection(spotlight_reshaped_data,
↪  reshaped_spotlight_range_grid, fastTime, prf, speed,
↪  cross_range_resolution, fc, fs);
backproject_image_stripmap = backprojection(spotlight_reshaped_data,
↪  reshaped_spotlight_range_grid, fastTime, prf, speed,
↪  cross_range_resolution, fc, fs);

figure(7)
imagesc((abs(backproject_image_spotlight)))
title('Spotlight SAR Data focused using back projection algorithm')
xlabel('Cross-Range Samples')
ylabel('Range Samples')

figure(8)
imagesc((abs(backproject_image_stripmap)))
title('Spotlight SAR Data focused using back projection algorithm')
xlabel('Cross-Range Samples')
ylabel('Range Samples')
```