Digital elevation modelling by radargrammetry in real-beam ground mapping mode

Ola Runeson

June 30, 2015

Abstract

In this thesis the problem of estimating terrain elevation using two-dimensional radar data from the multirole fighter aircraft JAS 39 Gripen is considered. Radar data contains information about range from the aircraft to the reflecting terrain, as well as horizontal angle. In general, radar data has high resolution in range and low resolution in angle, giving rise to interesting problems. A new radar with higher resolution is in development for the next-generation Gripen. This thesis aims at answering whether estimation of terrain height can be done using available radar data, in order to evaluate the plausibility of doing this with data from the new radar. The approach is to find matching terrain features in subsequent images of the ground, and use this information to calculate terrain elevation. Two approaches are implemented and studied, both on simulated radar data and on real datasets. One approach uses Harris corner detection and the other uses Speeded-Up Robust Features (SURF). Conclusions drawn are that the algorithms do not work for the available radar data, but that they possibly could work when higher resolution data from the new radar is available.

This page intentionally left blank.

Preface

This report is my thesis for the degree of Master of Science in Engineering Mathematics at Lund Institute of Technology (LTH).

The thesis work was conducted during 2015 at the aerospace and defense company Saab AB, business area Aeronautics, in Linköping. The company was founded in 1937 as the aircraft manufacturer Svenska Aeroplanaktiebolaget, but nowadays has a much wider range, including submarines, electronic defense systems, camouflage, weapons and logistics. Saab has produced many successful fighter aircraft such as Tunnan, Lansen, Draken, Viggen and Gripen.

Technical data such as radar resolution and ranges are classified. Therefore some results and data are omitted from the report. In these cases downgraded or fragmentary versions of the data will be included instead. It will be clearly stated where this has been done. No real radar data can be shown. A classified version of this report containing all information has been issued to Saab internally.

All work was done by me alone, with help from my supervisors at Saab: Ulf Rajgård and Alexander Rajula. I also recieved feedback on the writing of this report from Torleif Martin, my supervisor at LTH. Furthermore, Zoran Sjanic at Saab provided me with valuable code and discussions about keypoint tracking and height estimation. Roland Ericksson has also been helpful, answering questions about radar fundamentals, interpretations of phenomena in the radar data, and assessing which information is safe to disclose in the unclassified version of this report. Furthermore, David Johansson helped me find radar data from previously conducted test flights, and also let me fly the Gripen SYSIM simulator.

I would like to thank them all for their support and guidance that made this master thesis possible. Furthermore, I would also like to thank Per Hedberg for hiring me to do this master thesis at Saab, and everyone on my floor for contributing to a friendly and welcoming atmosphere.

Note: If you are reading the pdf version of this report, coloured numbers, words and sentences are clickable links to other parts of the document.

This page intentionally left blank.

Contents

Acronyms

AESA Active Electronically Scanned Array

DEM Digital Elevation Model

GRS80 Geodetic Reference System 1980

RBGM Real-Beam Ground Mapping

RLS Recursive Least Squares

RMSE Root Mean Square Error

SAR Synthetic Aperture Radar

SIFT Scale-Invariant Feature Transform

SURF Speeded-Up Robust Features

SWEREF99 Swedish Reference Frame 1999

SWEREF99TM Swedish Reference Frame 1999 Transverse Mercator

WGS84 World Geodetic System 1984

1 Introduction

The multirole fighter aircraft JAS 39 Gripen E (also known as Gripen NG), currently in development at Saab, will be equipped with a so called Active Electronically Scanned Array (AESA) radar. This type of radar forms its radar beam by shifting the phase of an array of transmitters/receivers as opposed to a mechanically scanned antenna, which physically moves the array. Aircraft radars often have a so called Real-Beam Ground Mapping (RBGM) mode, where they scan the ground in front of the aircraft and continuously display terrain features to the pilot. It is usually used for the purpose of aiding navigation, but this thesis evaluates its usefulness in creating a Digital Elevation Model (DEM) of the terrain, a kind of three-dimensional map. This is done by radargrammetry, a means of extracting geometric information from radar images. Since this modern AESA radar has high spatial resolution, creating DEMs of sufficient quality is plausible.

1.1 Background

When generating DEMs from radar data, a technique called Synthetic Aperture Radar (SAR) is often used. In SAR mode, a terrain area is selected and the radar beam continuously illuminates the area as the aircraft moves along some path. The measurements are then combined into a radar image of very high resolution. More information about SAR imaging is available in [1]. A short explanation of how these images can be used for radargrammetry is found in [2].

In the setting considered in this thesis, the radar will be in RBGM mode, meaning the beam will scan the ground from side to side. This results in radar images of significantly lower resolution, with the advantage of not having to focus on a specific area, but instead constantly monitor the terrain ahead of the aircraft.

The advantage of estimating a DEM from these images is that it can be generated while flying for some other purpose than mapping. Perhaps it could be used for displaying three-dimensional information to the pilot while in RBGM mode, aiding navigation. Furthermore, a continuous map of a long stretch of ground can be generated, as opposed to SAR, where for the two considered radars a specific patch of terrain has to be selected. The disadvantage is that DEM estimation will be harder because of the low resolution. Judging from published articles and literature, DEM estimation has not before been attempted using RBGM data.

It is hard to find open research about radar technology since there is usually a lot of secrecy involved, because of the apparent military applications. However, information about SAR imaging is easier to find since it has many applications in non-military remote sensing, such as environmental monitoring [3]. There is also a lot of research available in medical ultrasound imaging, where the methods used are very similar to those of real-beam radar imaging.

1.2 Thesis aim

The available radar data comes from the predecessor of Gripen E, the Gripen C/D, and has lower resolution. The aim of the thesis is to evaluate whether it is plausible or not to create DEMs of sufficient quality with the new AESA radar, by working with data from the old radar. The thesis aims at answering the following questions:

- Is it possible to create DEMs using the available RBGM data from the old radar?
- Which demands are there on data resolution and aircraft behaviour in order to obtain DEMs of sufficient precision? Is it plausible to use data from the new radar for this purpose?

1.3 Report structure

In general, the algorithm consists of a few consecutive steps. First, the radar data must be converted from polar to Cartesian coordinates and interpolated, in a process called scan conversion, which will be explained in more detail later. After this, noise in the image will have to be suppressed in order to make the following step easier. When this is done, the terrain height in the image will be estimated by radargrammetry, which is done by finding matching keypoints in consecutive images and then triangulating them.

First, some information about the hardware and radar specific phenomena will be presented. After that, more in-depth descriptions of the different problems involved are given along with a review of some available methods for solving them. Then comes a presentation of the available datasets. Thereafter comes a section about the methodology when developing and evaluating the algorithms. After this, the chosen methods and their implementations will be presented, followed by some experiments and results. Then a discussion of the results and a conclusion of the thesis will follow. At the very end, suggestions for future work on this project will be given.

1.4 History of radar technology

A historical description of the birth of radar is given in [4]. It is briefly summarized in this section.

In 1864, Maxwell published his theory about the existence of electromagnetic waves created by oscillating electric currents [5]. The existence of radio waves was discovered experimentally by Hertz in 1887, who published his results in [6]. This was seen as a verification of Maxwell's theory. In the year 1900, Tesla suggested a system that would use reflected radio waves to locate objects and measure the distance to them. Unfortunately, the idea was never funded and was eventually forgotten.

In the beginning of the 1900s there was no efficient way to amplify the weak reflected waves picked up by the receiver, but in 1904 the invention of the electron tube

provided this. In the same year, the German inventor Hülsmeyer invented the telemobiloscope [7], which could detect a ship at a distance of 5 km for the purpose of collision avoidance. However, he did not manage to sell the invention to anyone, so he abandoned it.

Coming up on the second world war, both the Germans and the British separately rediscovered these ideas. In Germany, the interest was mostly towards bomber targeting radars and ship detection radars, while the Brits were exclusively interested in air defense. Along the British coast, the Chain Home radars were built. These huge stationary antennas constantly transmitted radio waves in a 100 degree sector, and could only detect the range to the reflecting aircraft, not the direction. Ranges from different stations were then used to triangulate the aircraft positions. The German radars used rotating antennas which transmitted waves on shorter wavelengths, that could measure both range and angle to the target. Radar systems had a large impact on the outcome of the war in the European theatre.

Nowadays, radar technology is used for numerous purposes. According to [8], some applications are:

- Aircraft and ship navigation at night or in bad weather.
- Measuring altitude for aircraft or missile guidance.
- Detecting and measuring various weather phenomena.
- Detecting, locating and classifying aircraft, ships and spacecraft for defensive purposes.
- Mapping land, sea and ice.
- Detecting moving vehicles from the air.
- Targeting systems for guns and other weapons.
- Measuring distance and velocity for spacecraft navigation and docking.
- Precisely measuring astronomical bodies and their movements.
- Distance measurements for land surveying.
- Detecting and measuring underground objects, for example in archaeology.

2 Hardware

The considered radars measure luminance, which describes the amount of reflected radiation to the radar from a surface area of one square meter. An azimuth angle is an angle along the horizon, which in the considered radars is measured clockwise, i.e. a positive angle is to the right. For a specific azimuth angle, a train of radiowave or microwave pulses is transmitted along the radar beam, and the radar listens for the reflected pulses. Using the time from transmission of the received pulses, the so called slant range can be calculated. Slant range is the distance from the radar to the reflecting object along the radar line of sight. The pulses can be distinguished from each other by many different methods, for example by modulating the wave frequency or the time interval between pulses. The output from the radar is the measured luminance for each slant range and azimuth angle. This signal is called the radar video signal. A more thorough explanation of the principles behind radar systems can be found in [9].

In RBGM mode the radar beam is swept from side to side towards the ground in front of the aircraft. To cover enough ground area in each sweep, while simultaneously maintaining highest possible azimuth resolution, a so called fan beam is used. A fan beam is narrow along the horizontal direction and wide along the vertical direction, as can be seen in figure 1. The angular resolution is better when using a narrower beam, but there are both practical and physical limitations to how narrow beams can be achieved [10]. Radar resolution is generally high in slant range and low in azimuth.

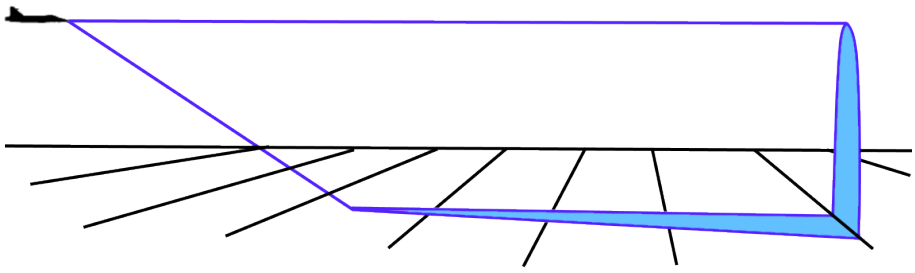


Figure 1: Example of tall and narrow fan beam used for RBGM.

2.1 PS-05/A

The PS-05/A is a radar initially developed by the electronics company Ericsson, but now by Saab, which is carried by Gripen versions A/B and C/D. Different versions of the radar has been in use since 1992. It is a mechanically scanned pulse-doppler radar, which means that it determines the slant range using different pulse timing along with measurements of the doppler shift, the change in frequency from transmission to reception. According to [11], the radar operates in the X-band, which means that it uses microwaves with frequencies in the range 8-12 GHz. The radar system weighs 150 kg and has a peak power of more than 10 kW. There are 13

available radar modes, including long range detection, multiple target tracking while searching for new targets, missile guidance, electronic counter-measure immunity, ground target identification, ground target tracking, air-to-ground ranging for bomb guidance, weather mapping, sea surface target searching and RBGM.

2.2 Raven ES-05

The Raven ES-05 is an AESA radar being developed by Selex ES in close cooperation with Saab, which will be carried by Gripen E. Just as the PS-05/A, it operates in the X-band. This AESA radar has many advantages over radars with mechanically scanned antennas. According to [12], it has significantly higher performance compared to mechanically scanned radars of the same weight and size. It has a wider field of view, 200 degrees from left to right. It can also switch the beam direction instantly, which makes it able to track multiple targets with more precision, and for example monitor the ground and airspace simultaneously. The radar weighs 215 kg and has 20 different radar modes, including RBGM.

In [13] it is stated that on the Gripen E, it is mounted on a so called swashplate, which can be seen in figure 2. When the radar looks straight ahead, it is displaced in some direction as compared to the aircraft, giving an even wider look angle in that direction. The swashplate can rotate very quickly to allow the radar to look equally far out in other directions, giving it a wider field of view.

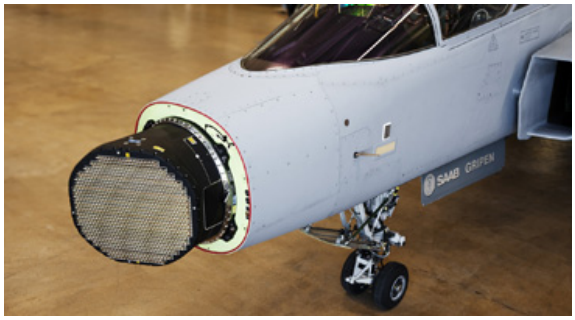


Figure 2: The Raven ES-05 radar mounted on a swashplate on the Gripen E. Figure source: [13]

2.3 Radar imaging phenomena

In radar images there are some phenomena present which are not intuitive in the same way as in a photographic image. These are described in [14] and summarized below.

2.3.1 Slant range scale distortion

Since the radar creates an image of the ground by measuring distances in slant range, there is always a scale distortion. Objects closer to the aircraft will always appear

smaller than objects farther away. In figure 3a the objects A1 and B1 have the same widths in reality, but their widths in slant range, A2 and B2, are different. This effect can easily be removed using simple trigonometry if aircraft altitude is known, but since the radargrammetry algorithm considered in this thesis works with slant range measurements, this is unnecessary.

In figure 3b, a SAR image affected by slant range scale distortion is shown. The fields to the left in the image are obviously compressed, but in the rectified image seen in figure 3c they are correctly represented.

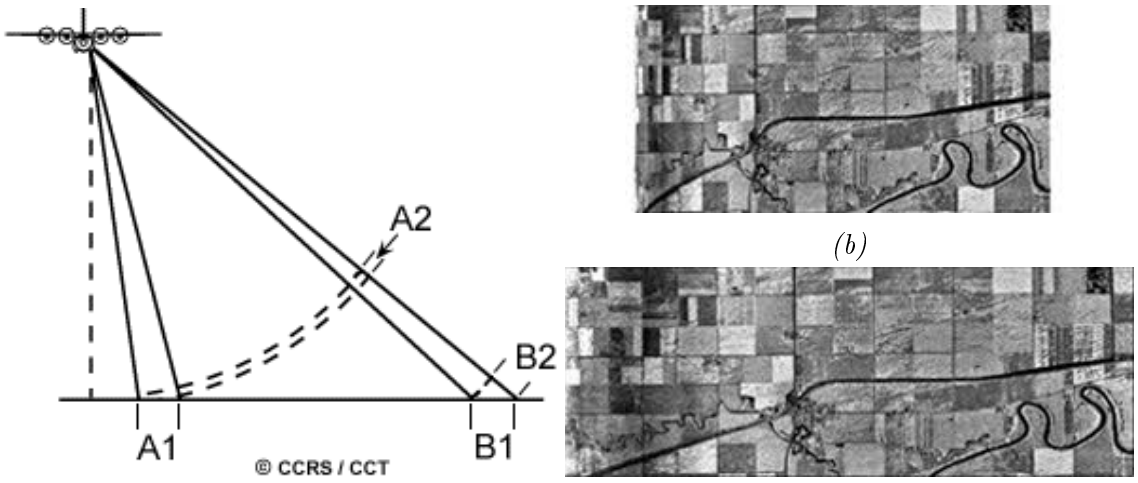


Figure 3: (a) Slant range scale distortion occurs when points at the same ground distance from each other are scaled differently depending on their distance from the radar. (b) A SAR image affected by slant range scale distortion, note the compressed fields in the leftmost parts of the image. (c) A rectified version of the same image where no distortion is present. Figure source: [14]

2.3.2 Radar shadow

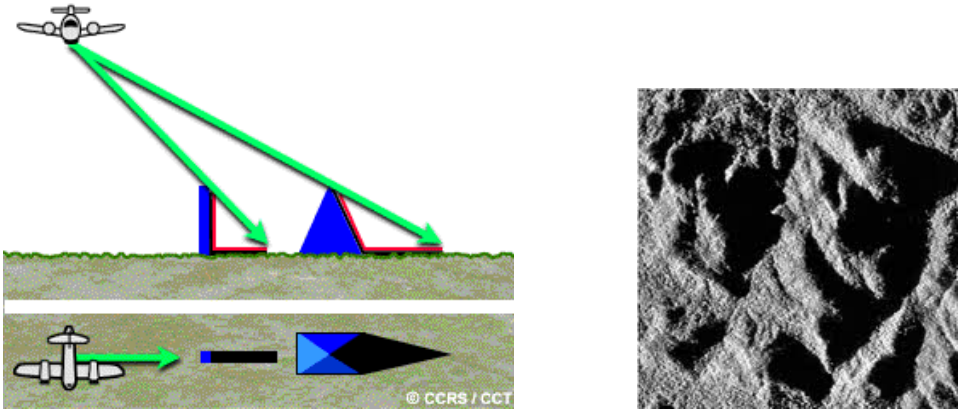
When the area behind a tall feature is not illuminated by the radar beam, an effect called radar shadow arises. It is very similar to shadows in the optical spectrum, except that they are caused by lack of irradiation from the direction of the radar itself, as opposed to an external light source. This effect is most apparent in steep terrain or when the aircraft is flying at low altitude.

Figure 4a explains the process graphically and a SAR image heavily affected by radar shadow is seen in figure 4b.

2.3.3 Foreshortening

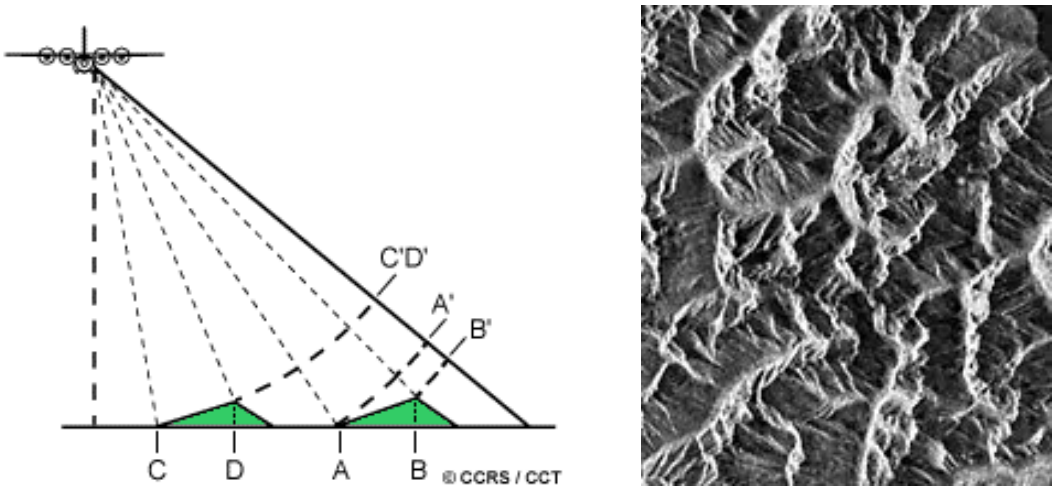
When the terrain is sloping towards the radar, an effect called foreshortening arises. The two points A and B in figure 5a appear closer to each other in the radar image

than they are in reality. The effect is most severe when the slope is perpendicular to the radar beam so that the whole slope is mapped to the same slant range, as for the points C and D. The amount of foreshortening in an image depends on the terrain slope in relation to the altitude of the aircraft. A SAR image with severe foreshortening can be seen in figure 5b, where the terrain sloping towards the radar is compressed and appears extra bright. This effect can not be rectified without prior knowledge of the terrain.



(a) *(b)*

Figure 4: (a) Radar shadows occur when the area behind an obstacle is not illuminated by the radar beam. (b) A SAR image that has severe radar shadows on the far side of the mountains. Figure source: [14]

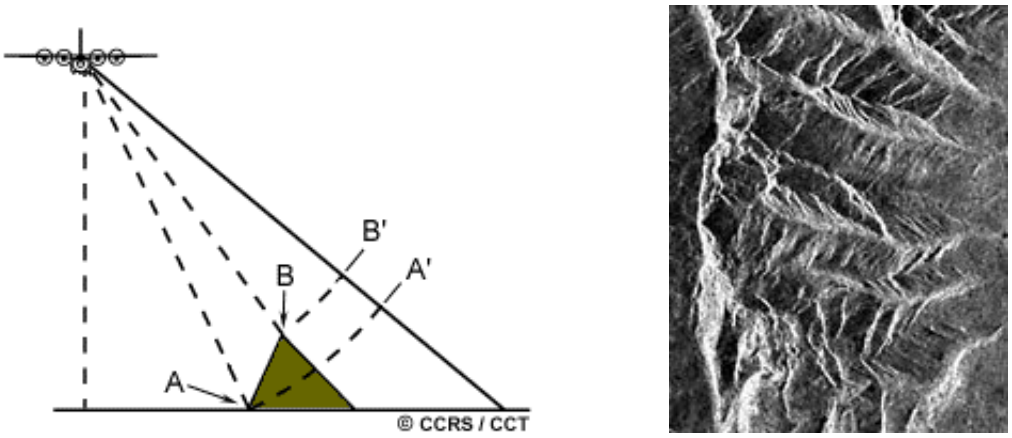


(a) *(b)*

Figure 5: (a) Foreshortening occurs when hillsides sloping towards the radar are compressed in slant range as compared to actual ground distance. (b) A SAR image that shows the effects of foreshortening. Figure source: [14]

2.3.4 Layover

When the slopes are so steep that the radar beam reaches the top before it reaches the bottom, layover occurs. This results in an effect where top and bottom show up in the wrong order, as can be seen in figure 6a. In an image this effect looks as if the top is rolling towards the radar, like a crashing wave. A SAR image suffering from this effect is shown in figure 6b. This effect can not be rectified without prior knowledge of the terrain.



(a) *(b)*

Figure 6: (a) Layover occurs when slopes are so steep that the order of points in slant range is not the same as the order along the ground. (b) A SAR image that shows the effects of layover. Figure source: [14]

2.3.5 Sea clutter

When the radar is pointing at the sea in a direction perpendicular to the waves, sea clutter occurs. It is caused by radar reflections from wave crests. In the PS-05 radar imagery it looks like smoke, an example can be seen in figure 7. Since the figure depicts water, the intensity should be homogeneous.

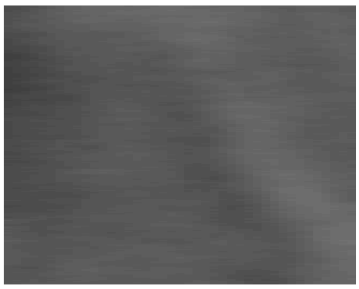


Figure 7: A cutout from radar data showing sea clutter. The scale has intentionally been left out.

3 Theory

3.1 Geography

In order to interpret navigation data and relate measurements to the real world, some knowledge of geography is needed.

3.1.1 Earth models

3.1.1.1 Ellipsoid

The earth is usually portrayed as a sphere, but in reality this is not true [15]. The earth resembles the volume obtained by rotating an ellipse around the polar axis. This shape looks like a sphere that has been flattened at the poles and is commonly referred to as an ellipsoid. An ellipsoid is fully defined by its major and minor semiaxes, also referred to as the equatorial radius and the polar radius. It can also be equivalently defined by its semimajor axis and its flattening constant, which is a constant related to the curvature of the surface. There are many different ellipsoid models of the earth and the one used in this thesis is called Geodetic Reference System 1980 (GRS80) which was estimated from satellite measurements in 1980. It is very similar to the World Geodetic System 1984 (WGS84) ellipsoid, that is used in many GPS systems.

According to Lantmäteriet [16], the GRS80 ellipsoid has a semimajor axis of 6 378 137 meters and a flattening constant of 1/298.257222101.

3.1.1.2 Geoid

As stated above, the ellipsoid is an approximation of the earth. There are better models available, describing the deviation from a reference ellipsoid, such as GRS80. A geoid is a model which takes into account the distribution of land mass over the ellipsoid surface [15]. It follows the gravitational field of the earth, being higher in areas where there is more land. The shape of the geoid varies at most up to about 100 meters from the ellipsoid. Terrain elevation is usually specified as height over the geoid, commonly referred to as height above sea level. This is because higher gravity attracts more water, which gives a higher theoretical sea level.

The geoid model used in this thesis is made by Lantmäteriet and was developed specifically for Sweden. It is called Rikets Höjd 2000, and it is described in [17]. An exaggerated geoid model of the earth is presented in figure 8.

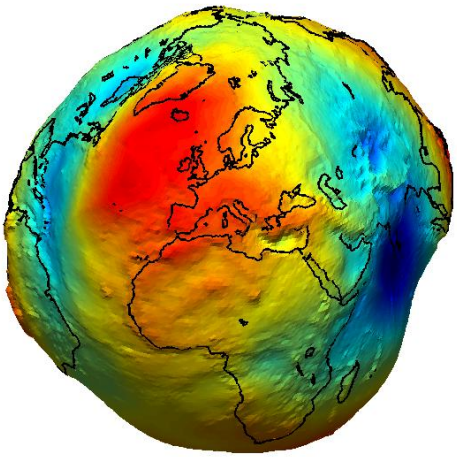
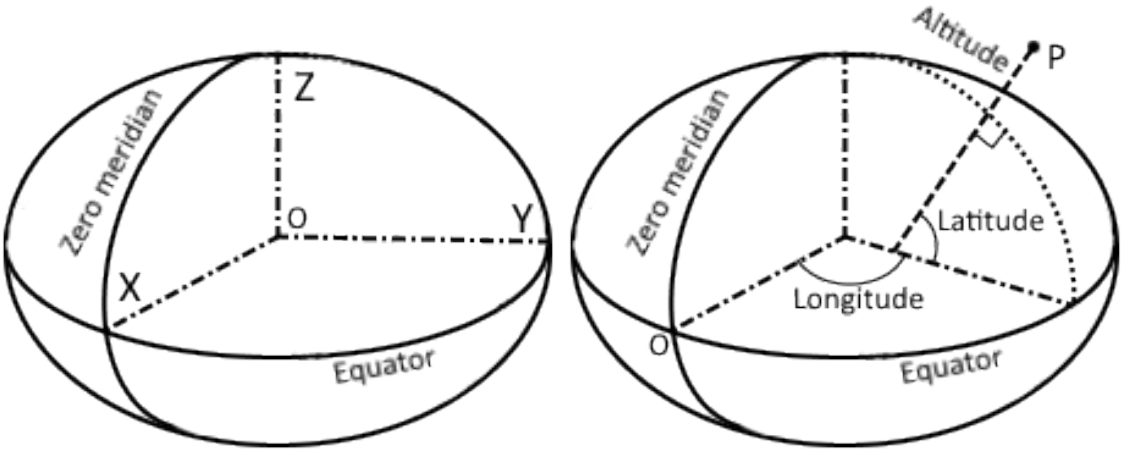


Figure 8: An exaggerated geoid model of the earth. Figure source: [18]

3.1.2 Coordinate systems

3.1.2.1 Swedish Reference Frame 1999

SWEREF99, is a three-dimensional cartesian coordinate system, defined by Lantmäteriet [19], where points are given on the form $p = (x, y, z)$, illustrated in figure 9a. The unit for each of these coordinates is meters and the origin is situated at the center of the earth. The x-axis points towards the intersection of the zero meridian and the equator, the z-axis points towards the north pole and the y-axis points 90 degrees east, so that they form a right hand oriented system.



(a) *(b)*

Figure 9: (a) Illustration of the SWEREF99 coordinate system. (b) Illustration of the geodetic coordinate system.

3.1.2.2 Geodetic coordinates

Another way of describing points on or above the surface of the earth, is to use the so called geodetic coordinates: latitude, longitude and altitude. These are related to a specific ellipsoid, in the context of this thesis the GRS80 ellipsoid defined above. According to [15], longitude is defined as degrees along the equator to the east or west of the zero meridian, and thus range between 180 degrees east and 180 degrees west. This is often simplified by defining east as positive direction, which makes longitude range from -180 to 180 degrees. Latitude is defined in an analogous manner as degrees north or south of the equator, which is simplified to a range of -90 to 90 degrees. This angle is measured between the equatorial plane and the normal, not at the center of the earth. Altitude is defined as meters above the ellipsoid along the surface normal. This coordinate system is illustrated in figure 9b.

Converting from SWEREF99 to geodetic coordinates can be done analytically, but this involves solving fourth degree equations which is hard in practice. There are several iterative algorithms approximating the solution to those equations.

According to Ligas and Banasik [20], the longitude λ in radians can be found by the simple calculation

$$\lambda = \begin{cases} \frac{\pi}{2} - 2\arctan\left(\frac{x}{\sqrt{x^2+y^2}+y}\right), & y \geq 0 \\ 2\arctan\left(\frac{x}{\sqrt{x^2+y^2}-y}\right) - \frac{\pi}{2}, & y < 0 \end{cases}.$$

The latitude and altitude is a harder problem to solve. According to Fukushima [21], it can be approximated by running his iterative algorithm for only one iteration. This gives a maximum error in latitude of 2 milliarcseconds, which at worst corresponds to only about 0.6 meters at relevant altitudes. An implementation of this algorithm is be described in section 6.1.1.

3.1.2.3 Swedish Reference Frame 1999 Transverse Mercator

SWEREF99TM is a two-dimensional coordinate system for the surface of the earth, customized for optimal accuracy in Swedish territory. It is defined by Lantmäteriet and described in [22]. Points in this reference frame are projections of their three-dimensional representations onto the GRS80 ellipsoid. The two coordinates are called northing and easting. Northing means meters north of the equator, and easting is meters east of a center meridian, both measured along the ground. The projection used in SWEREF99TM is called Gauss' conformal projection, or Transverse Mercator projection. It uses the 15 degree meridian as center and employs a scale factor of 0.996 to account for the latitude of Sweden. Furthermore, 500 000 meters are added to the easting values in order to get approximate correspondence with the Universal Transverse Mercator projection for the whole world.

An implementation of this algorithm is presented in section 6.1.2.

3.2 Scan conversion

The radar video signal is delivered in so called spokes, where each spoke corresponds to a specific azimuth angle. The spoke contains information about the measured luminance from each slant range along that angle, divided into discrete slant range bins. It also contains navigation data that specifies the aircraft position from which each spoke was recorded. These spokes are indexed into frames, where each frame contains all spokes in one scan from side to side. Thus, each pixel in a frame corresponds the measured luminance in a specific azimuth angle, and a specific interval of slant ranges. An illustration can be seen in figure 10.

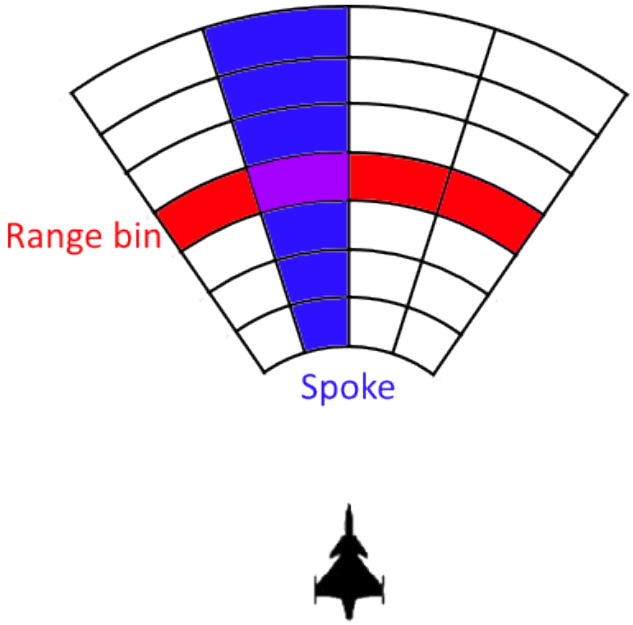


Figure 10: Explanation of the notion of spokes and range bins.

Since the radar video signal is delivered in spokes and subsequently collected into frames in polar coordinates, it needs to be converted into Cartesian coordinates for visual interpretation. The relation between these coordinate systems is visualized in figure 11a.

Since the spokes correspond to discrete azimuth angles there will be holes in the Cartesian output image, that need to be filled. This phenomenon is illustraded in figure 11b. The angular resolution can be increased by sampling at closer spaced angles, but as stated previously the resolution is physically bounded by the radar. The holes can be filled by two different strategies, described below. In each of these strategies an interpolation scheme has to be chosen. Different choices will also be described below.

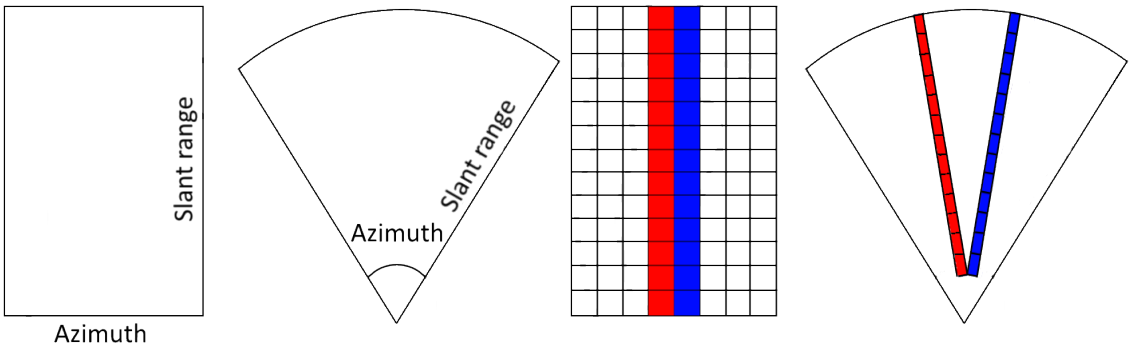


Figure 11: (a) Scan conversion is a transformation from a frame in polar coordinates (left) to Cartesian coordinates (right). (b) Since the spokes in polar coordinates are sampled at discrete azimuth angles there will be holes in the converted data that increase with slant range.

3.2.1 Forward scan conversion

In forward scan conversion, every pixel in the azimuth-range plane is converted into Cartesian coordinates in the output image using simple trigonometry. For given azimuth angle ϕ and slant range r, the mapping onto Cartesian coordinates is (see figure 11a)

$$x = r \sin \phi \tag{1}$$
$$y = r \cos \phi. \tag{2}$$

After this conversion, the holes are filled by interpolation in the x-y plane. This interpolation has to be done in an angular fashion, not along the x and y axes.

3.2.2 Reverse scan conversion

In reverse scan conversion every pixel (x, y) in the output image is mapped to its corresponding value in the ϕ-r plane. Solving (1) and (2) for ϕ and r gives the inverse mapping

$$\phi = \arctan(y/x)$$
$$r = \sqrt{x^2 + y^2}$$

The advantage of this strategy is that the mapping will not have to be computed for all pixels in the radar video signal, just the ones that are needed. The pixel value is then calculated by interpolation in the ϕ-r plane, which means it can be computed along each basis direction (ϕ and r) instead of in an angular fashion, as in forward scan conversion [23].

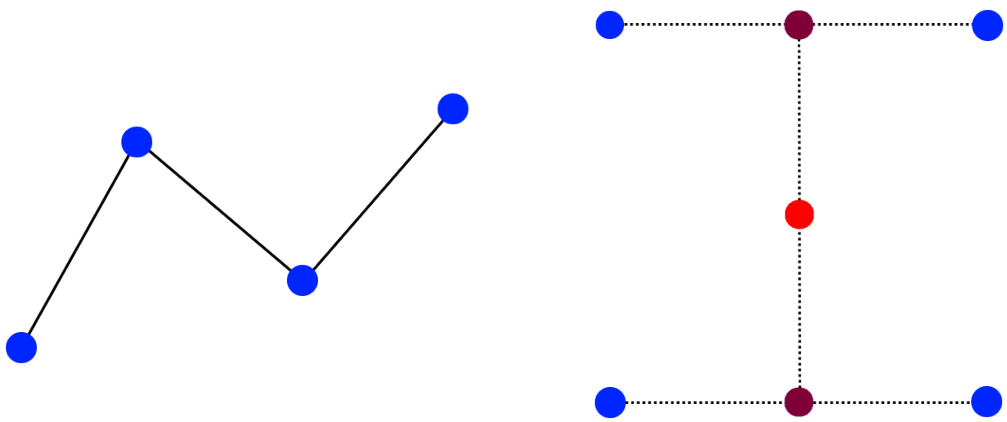
3.2.3 Interpolation

The following two interpolation schemes are considered.

3.2.3.1 Bilinear

Bilinear interpolation is a generalization to two dimensions of the well-known linear interpolation method. To find the function value at a location between existing data points, a piecewise linear function which passes through the data points is constructed. The interpolated value is then picked as the function value at the sought location. The interpolated function is guaranteed to be continuous, an example can be seen in figure 12a. To find the value at a given point, only the two neighbouring data points need to be considered.

The generalization to two dimensions is straightforward. To find the interpolated value at a given point, the four closest data points are needed. Linear interpolation is done along one direction, to find two intermediate points, and then linear interpolation between these intermediate points is done in the other direction. This process is illustrated in figure 12b.



(a) *(b)*

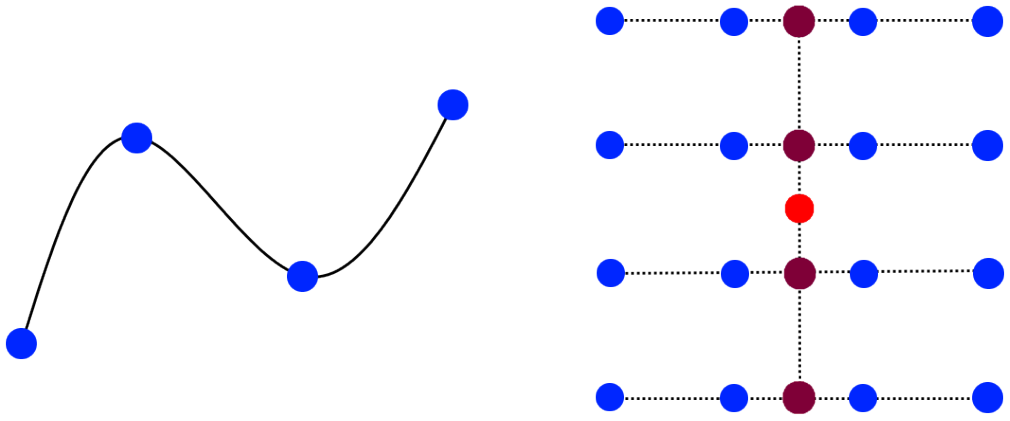
Figure 12: (a) Example of a piecewise linear function passing through the data points. (b) Clarification of how linear interpolation is generalized to two dimensions. Data points are blue, intermediate points are purple and the sought point is red.

3.2.3.2 Bicubic

Bicubic interpolation is a generalization to two dimensions of the cubic interpolation method. To find the function value at a location between existing data points, a piecewise cubic function which passes through the data points is constructed. The derivatives at the end points of each cubic interval are required to be equal. The interpolated value is then picked, in the same manner as above, as the function value at the sought location. The interpolated function is guaranteed to be smooth,

which means that it is continuous with continuous derivative. An example can be seen in figure 13a. To find the value at a given point, the four closest data points are needed.

The generalization to two dimensions is analogous to what was done in the bilinear case. To find the interpolated value at a given point, the 16 closest data points are needed. Cubic interpolation is done along one direction, to find four intermediate points, and then cubic interpolation between these four intermediate points is done in the other direction. This process is illustrated in figure 13b.



(a) *(b)*

Figure 13: (a) Example of a cubic function passing through the data points. (b) Clarification of how cubic interpolation is generalized to two dimensions. Data points are blue, intermediate points are purple and the sought point is red.

3.3 Denoising

The radar images are affected by different types of noise. This noise has to be suppressed before keypoint tracking, since noise varies from frame to frame, which makes corresponding points harder to match.

3.3.1 Noise sources

When radar waves travel through the atmosphere they are affected by absorption and scattering [10]. Absorption is mostly due to oxygen and water vapour, and scattering is almost exclusively due to condensed water vapour. When absorption occurs, the energy of the received echo will be lowered, resulting in lower luminance measurements. Scattering means that some of the radiation will be scattered in many directions when hitting particles of sizes comparable to the wavelength, such as water droplets or hail. Snow flakes and clouds scatter radiation much less. Smoke and dust has negligible effect because of their small particle sizes. This means that the echo will have lower energy, but if enough radiation is scattered back in the direction of the receiver it can even increase the luminance measurement. These two

effects combined are called atmospheric noise, which gets worse at higher frequencies. The noise power decreases with lower humidity.

External electrical noise, from other sources than the radar itself, is high for radars using low frequencies [10]. This noise is called galactic noise and the main sources are the sun, the atmosphere and the ground. Sun radiation varies with solar conditions, while earth and atmosphere radiation varies with temperature. On the ground, not only temperature affects the emitted radiation, but also the material. Since water is a good conductor, a body of water emits considerably less radiation than a piece of land with the same temperature. The received radiation from the atmosphere also depends on how much atmosphere the radar line of sight is intersecting. Furthermore, the sun radiation will be higher if the radar is looking towards the sun. These noises all decrease with higher radar frequency. In the X-band, where both of the two considered radars operate, atmospheric noise is the most powerful of the two noise sources mentioned above.

The most powerful source of noise is the receiver itself [10]. Since the movement of free electrons is random due to thermal agitation, at any given instant more electrons in the receiver will be moving in one direction than in others. This movement causes a voltage that will disturb the luminance measurements. This voltage increases with receiver temperature.

3.3.2 Denoising filters

A spatial filter is applied in such a way that for every pixel in the image, it considers the values of all pixels in a certain neighbourhood. From this information it decides the pixel value in the output image. The neighbourhood is in most cases a rectangular window around the considered pixel. In the case of noise suppression, the filter will in some way average the pixel values in the window, either by mean, mode, median or more advanced methods, such as the sigma filter described below.

Mean and mode filters are well-known to blur and distort edges in images [24], and are therefore not considered in this thesis.

3.3.2.1 Median filter

A median filter, as the name implies, sets the value of the center pixel to the median of the pixels in the surrounding window. It is better at preserving edges than the mean and mode filters [24], since for a pixel close to an edge the window will mostly contain pixels on the same side, see figure 14. This means the median will be picked among the values from that side, preserving the edge. However, it has a tendency to erode sharp corners in images, since a majority of the pixels in the window will belong to the region outside the corner, as can be seen in the same figure. The median filter is very easy to implement and computationally efficient.

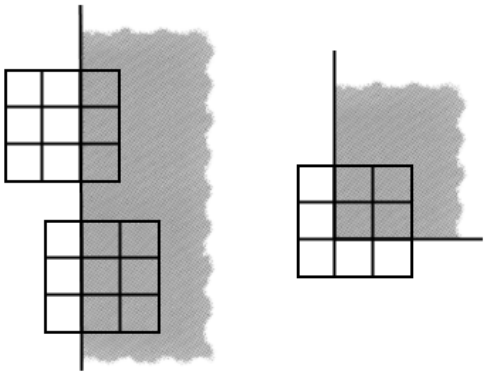


Figure 14: A median filter picks the value of the center pixel from the correct side of an edge, since most pixels in the window will be on that side. For a corner, most pixels in the window will belong to the wrong region.

3.3.2.2 Sigma filter

In [25], Lee proposes a filter that is better than the median filter at suppressing noise while preserving edges. It works by forming the mean of all pixels in the window that are within a given intensity range from the center pixel. According to Lee this filter is computationally efficient.

The 2σ probability is the probability that a sample from a random variable is within two standard deviations from its mean. As an example, the common Gaussian distribution has a 2σ probability of 95.5%. If the pixel value is modelled as the real luminance measurement with noise, the sigma filter considers a pixel to belong to a different region in the image if its value is outside the 2σ range of the central pixel.

For each pixel in the image, with intensity x, the sigma filter does the following:

1. Calculate an intensity range $[x - 2\sigma, x + 2\sigma]$ around the pixel value.
2. Select all pixels in the current window that fall within the intensity range.
3. Set the output pixel value to the mean of the selected pixels.

To be able to also suppress noise where a few neighbouring pixels have similar values, the following modification to the last step in the above algorithm is made:

3. If the number of selected pixels is less than K, set the output pixel value to the mean of the immediate neighbours (3×3 window). Otherwise use the mean of the selected pixels.

This modification gives the biased sigma filter.

The parameters that have to be chosen are the window size, the noise standard deviation σ and the minimum number of selected pixels K. Window size should be chosen large enough to efficiently suppress noise, but small enough to retain a low execution time. The K parameter should be chosen large enough to effectively suppress noise, but small enough to not destroy detailed features in the image. Lee

recommends a K value less than 3 for a 5×5 window and less than 4 for a 7×7 window. These values ensure that a line with a width of one pixel will be preserved. The σ parameter can either be set to a suitable value, or estimated as the standard deviation of a region in the image assumed to be homogeneous. Furthermore, Lee states that it may be advantageous to filter the image iteratively a few times, in that case the number of iterations also has to be determined. In practice, these parameters should be chosen by experimentation to tune them to the considered type of image.

3.4 Radargrammetry

Radargrammetry is the process of finding matching keypoints in two or more radar images, and then using triangulation to estimate the height above a reference model of the earth, see section 3.1.1. It is basically the same process as estimating depth from a set of photographic images, so called photogrammetry. However, there are some radar specific phenomena that are not present in photographs, recall section 2.3. Furthermore, in photogrammetry we have access to both the horizontal and vertical angles to each point, but not the range, whereas in radar images we have access to range and horizontal angle, but not vertical angle.

3.4.1 Keypoint tracking

The problem of finding matching points in two images can be segmented into two subproblems. The first problem consists of finding interesting points, henceforth referred to as keypoints, in the images. The second problem is to determine which of these points match. There is not always a need for finding keypoints in the second image before searching for matches. Some methods find keypoints in the first image and then evaluate all points in a subset of the second image to find the best match. When one of these is done for a sequence of images, the matched keypoints will form tracks in the azimuth-range plane.

3.4.1.1 Harris corner detector with digital image correlation

An ubiquitous method for finding interesting points is the Harris corner detector, presented by Harris and Stephens in [26]. As the name implies, it searches for corners in an image. The basic principle is taking a patch around each pixel and translating it in different directions. If the patch is a flat part of the image, the translated versions will be quite similar to the original one. If an edge is present in the patch, the translated versions will differ when translated across the edge, and be quite similar when translated along the edge. However, if a corner is present, the translated versions will differ in many directions. This means that a detected corner is not a corner in the usual sense, but rather a point of fast variations that is not an edge.

Let the variables u, v iterate through all pixels in a weighted window w around a specific pixel and let x, y denote an arbitrary translation of this window. Furthermore, let $I(x, y)$ denote the intensity of a specific pixel in the image. Then the weighted sum of squared differences (similar to the notion of zero-mean normalized sum of squared differences in the section below) is

$$E(x, y) = \sum_{u,v} w(u, v) \left| I(u + x, v + y) - I(u, v) \right|^2 .$$

Doing a Taylor expansion of $I(u + x, v + y)$ around $(x, y) = (0, 0)$ gives

$$\begin{aligned} I(u + x, v + y) &\approx I(u, v) + \frac{\partial I}{\partial x}(u, v)x + \frac{\partial I}{\partial y}(u, v)y \\ \implies E(x, y) &\approx \sum_{u,v} w(u, v) \left| I(u, v) + \frac{\partial I}{\partial x}(u, v)x + \frac{\partial I}{\partial y}(u, v)y - I(u, v) \right|^2 \\ &= \sum_{u,v} w(u, v) \left| \frac{\partial I}{\partial x}(u, v)x + \frac{\partial I}{\partial y}(u, v)y \right|^2 . \end{aligned}$$

The latter is a quadratic function of x and y, which means it can be rewritten as

$$E(x, y) \approx \begin{bmatrix} x & y \end{bmatrix} A \begin{bmatrix} x \\ y \end{bmatrix},$$

with

$$A = \sum_{u,v} w(u, v) \begin{bmatrix} \left(\frac{\partial I}{\partial x}(u, v)\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}(u, v) \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y}(u, v) & \left(\frac{\partial I}{\partial y}(u, v)\right)^2 \end{bmatrix} .$$

Since this function is written on quadratic form, A is its Hessian with respect to x and y. The eigenvalues of the Hessian are proportional to the variation along the corresponding eigenvectors. The sum of squared differences, E, will be large if there are major changes resulting from a translation, and small if the patch is similar after translation. Recalling that we wanted large variation in all directions, a corner point is characterized by both eigenvalues being large.

After finding a keypoint in the first image, digital image correlation is a method of finding the best match among all points in a certain subset of the second image. Pan, Xie and Wang present different ways of performing this check and conclude mathematically and experimentally that they are equivalent [27]. The most computationally effective method that is robust is shown to be zero-mean normalized sum of squared differences, denoted by C_{ZNSSD}.

Given a keypoint in the first image, a patch f around it is selected to describe the point, a so called descriptor. In the second image a subset, called the search region, is specified, typically near the keypoint position in the previous image. For every pixel in the search region, a candidate patch g of the same size as f is selected. For every candidate patch g, the C_{ZNSSD} is calculated as

$$C_{\text{ZNSSD}} = \sum \left(\frac{\bar{f}_i}{\sqrt{\sum \bar{f}_i^2}} - \frac{\bar{g}_i}{\sqrt{\sum \bar{g}_i^2}} \right)^2 ,$$

where f_i, g_i denotes individual pixels in f and g. The variables $\bar{f}_i$ and $\bar{g}_i$ are the pixel intensities with the averages of f and g subtracted respectively. The sum is iterated over all available pixels in the descriptor and the candidate. The value of C_{ZNSSD} lies between 0 and 4, where 0 is a perfect match. After these calculations, the matching point is considered to be the candidate that has the lowest C_{ZNSSD} value. This matching method is claimed to be robust against drift in mean intensity between the images, as well as translation offset and scale changes [27].

3.4.1.2 Speeded-Up Robust Features

Speeded-Up Robust Features (SURF), is a faster relative of the well-known Scale-Invariant Feature Transform (SIFT) algorithm. It was proposed in 2006 by Bay, Tuytelaars and Gool. In 2009, it was clarified and implemented as OpenSURF by Evans [28]. SURF works by detecting keypoints in both the first and second image, calculating a descriptor for each keypoint, and then finding the best match using these descriptors.

To find keypoints in an image, a so called scale-space is constructed. It describes the image at different scales, which can be regarded as different levels of detail. In practice, one scale is constructed by low-pass filtering the image by convolution with a Gaussian, the next scale is constructed by convolution with a Gaussian of different size, and so on. For all pixel locations and all scales, the local Hessian is approximated. The determinant of the Hessian, called the discriminant, will be positive for extreme points in scale-space, where larger values mean more distinct peaks. Points where the discriminant has local maxima are considered keypoints. The main difference from SIFT is that SURF uses clever approximations and implementations to increase computational efficiency.

The descriptors are extracted in two steps. First an orientation has to be assigned to each descriptor to allow them to be rotationally invariant, but in this thesis that is not needed. Even without orientation assignment, SURF can match points even after up to 15 degrees rotation. The second step is to extract the descriptor components. This is done by forming a square window of size 20σ, where σ is the scale at which the keypoint was detected, and dividing it into 16 subregions. In each of these subregions four values are extracted. After doing this for every subregion, a descriptor vector of 64 components is obtained. The Euclidian distance between two descriptor vectors describes their similarity, with 0 being a perfect match.

3.4.2 Height estimation

The working principle for estimating the height of one of these keypoints, is that a range and an azimuth angle from the aircraft to the keypoint defines a circle in three-dimensional space. The terrain feature corresponding to the keypoint is located somewhere on this circle. Detecting the same keypoint in the next frame defines another circle, centered around the new aircraft position, and so on. With perfect measurements and matching, these circles would all intersect in one point,

which is the real three-dimensional location of the terrain feature. Since this will never be the case, an optimization algorithm has to find the best point of intersection between all circles, which gives the terrain feature's position. An illustration of this principle in two dimensions is shown in figure 15. It can be thought of as finding the height of a point that is straight ahead of the aircraft in all frames.

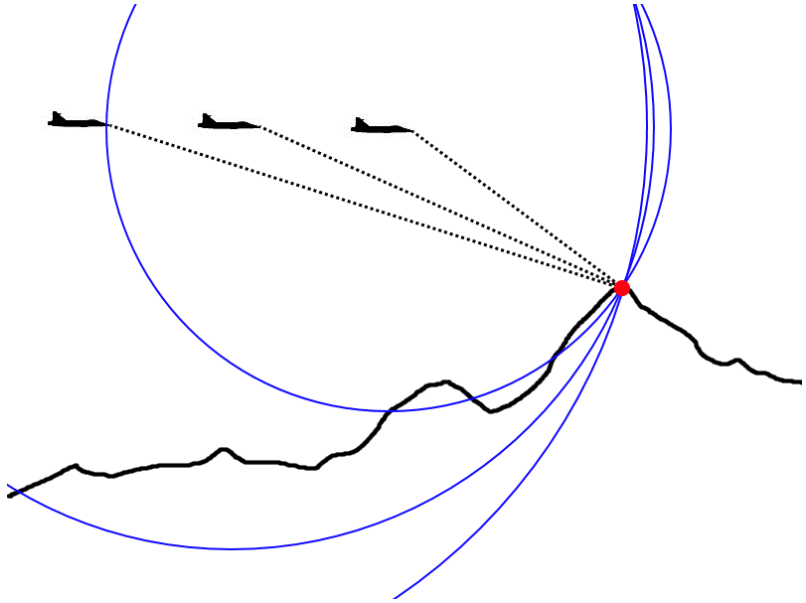


Figure 15: The principle of height estimation by approximating the intersection of many circles. The figure has been drawn in two dimensions for simplicity.

4 Datasets

4.1 DEM data

Data for result evaluation is available from the Swedish national land survey, Lantmäteriet. This data has been acquired free of charge according to an agreement between Lantmäteriet and Lund University, specified in [29].

The data comes from a DEM called *GSD-Höjddata, Grid 2+*, which almost covers all of Sweden. The coverage is illustrated in figure 16. The data quality is specified in [30]. It was obtained by airborne laser scanning between 2009 and 2015. It has a spatial resolution of 2×2 meters and a mean error of 5 centimeters. Generally, the height resolution is a lot better, but local errors in some regions increase the mean error. As can be seen in the shaded surface in figure 17, the resolution is fine enough to show small ditches in farm fields and trails in wooded areas.

Figure 16: Coverage of the GSD-Höjddata, Grid 2+ DEM from Lantmäteriet. The blue boxes indicate areas scanned during the same period of time.

The height information is given in meters above sea level according to the geoid, and will have to be converted to meters above the ellipsoid. The concepts of geoid and ellipsoid were explained in section 3.1.1. The conversion is trivial; since the geoid model is specified in meters above the ellipsoid, it is done by simple addition.

There are many error sources in this DEM, but this does not prevent its use as evaluation data in this thesis, since it will still be much more accurate than the DEMs created from low-resolution radar data. The main error sources are dense vegetation being classified as ground, very steep terrain that is hard to measure from above, and high water levels that cover the ground and thus distort shorelines.

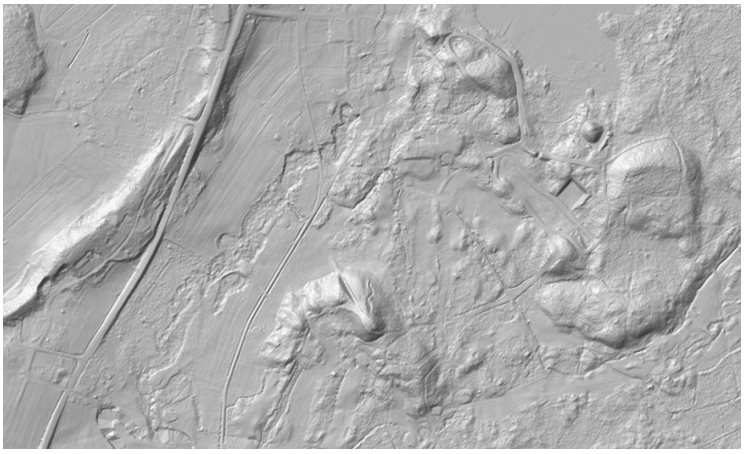


Figure 17: A shaded surface describing the fine height resolution in the laser scanned data. Figure source: [31]

4.1.1 Abisko dataset

In particular, an extracted piece of the DEM containing the town Abisko, the lake Torne träsk and a number of Swedish mountains is used in the experiments. The region has been chosen since it has both flat areas and very hilly areas, which makes visual result interpretation easier. This DEM is referred to as the Abisko dataset. It spans 25 km from east to west and 30 km from north to south. In altitude it ranges from 370 to 1760 meters above the ellipsoid. For the sake of computational efficiency it is downsampled by a factor of 200 as compared to the DEM described in the above section. A three-dimensional plot of the dataset is seen in figure 18.

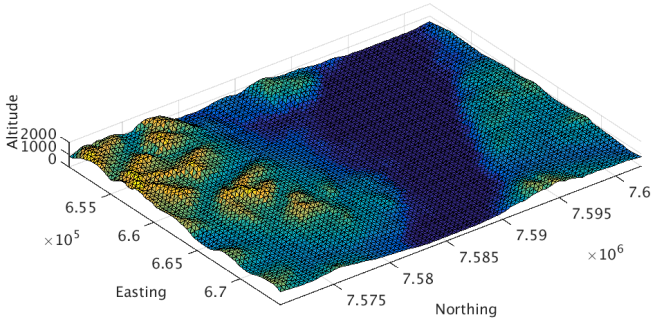


Figure 18: A three-dimensional plot of the Abisko dataset downsampled by a factor of 200.

4.2 Radar data

Finding suitable radar data is not an easy task. During most test flights the aircraft seems to turn and change altitude very often. Because of this, the available datasets do not include long segments where the same ground area is illuminated. However, three short but suitable datasets were extracted from longer test flights. These are described below, but unfortunately none of them can be included in this report since they are classified.

4.2.1 Östergötland dataset

The first set of radar data is data from Östergötland, where the aircraft approaches the coast from southeast. Both Linköping and Norrköping are clearly visible in the images. The data is collected from an altitude of about 5500 meters, with the radar pointing down at the surface close to the aircraft. This dataset is 61 frames long.

4.2.2 Stockholm dataset

In this dataset the aircraft flies northeast along the coast south of Stockholm at an initial altitude of 3000 meters and during the dataset climbing to 8000 meters. The aircraft flies perpendicular to the sea wave direction, resulting in severe sea clutter, as described in section 2.3.5. This dataset is 56 frames long.

4.2.3 Gotland dataset

In this dataset the aircraft flies southeast towards Gotland at an altitude of around 3000 meters. The northernmost part of Öland is visible in the right part of the radar data. Some sea clutter is visible, but not as severe as in the Stockholm dataset. This dataset is 31 frames long.

5 Methodology and metrics

5.1 Implementations and experiments

When handling navigation data, algorithms for converting between the geographical coordinate systems described in section 3.1.2 are needed. The implementations of these will be briefly described.

Since the radar data is usually processed by a program called Mission Support System, the binary radar data has to be decoded and stored in Matlab format according to internal documentation about the bit patterns. The data files are quite large, so they have to be stored and processed in a smart way. General aspects of the implementation of this data decoding will be described.

When the data is decoded, the reverse scan conversion algorithm described in section 3.2.2 is implemented, so that decoded radar data can be read from the hard drive and converted in real time. Since this is an integral part of all experiments regarding radar data, it has to be fast in order to make testing and validation as quick as possible. The implementation will be described and the results from a validation experiment will be presented. Furthermore, the two interpolation schemes described in section 3.2.3 will be compared.

After this, the two denoising filters described in section 3.3 are tested and evaluated, with the intent of selecting the most suitable alternative. Some information about their implementations will be presented, as well as the experiment results.

Then the height estimation algorithm is validated by selecting keypoints in the DEM from Lantmäteriet and simulating aircraft trajectories above it. Azimuth and slant range measurements are synthesised from these aircraft positions, initially noise free. After running the algorithm on these keypoint tracks, the results are evaluated in both quantitative and qualitative terms. This experiment is repeated for different numbers of keypoints, different track lengths and varying aircraft trajectories. This aims to validate the algorithm's functionality, find out how the parameters influence the output and give indications as to what kind of quality can be expected under perfect conditions. Some of the experiments are then repeated, with noise added to the azimuth and slant range measurements, with the intent of finding out how long tracks are required in order to obtain sufficient DEM quality. The implementation will be described and the experiment results, for both the noise free and the noisy case, will be presented.

When the height estimation algorithm is validated and tested, the two keypoint tracking algorithms described in section 3.4.1 will be implemented and tested on the available radar datasets. Studying the track lengths and considering the radar data resolution, an assessment of the plausibility of obtaining DEMs of sufficient quality is made, according to the results obtained from experimenting with the height estimation algorithm.

5.2 Performance metrics

When evaluating the results of each step in the algorithm, some performance metrics are needed. Quantitative measures are preferable, since these facilitate objective evaluations and comparisons.

5.2.1 Scan conversion

To evaluate the scan conversion, a qualitative assessment of the output as compared to raw data is needed, to see if they are reasonable. Comparing the two interpolation schemes is done qualitatively as well, but the execution times for scan conversion of each frame can be compared in a quantitative manner.

5.2.2 Denoising

It is hard to find objective measures of image processing results. Interesting properties are noise suppression and how well edges are preserved. The results will be qualitatively evaluated according to these factors, by looking at the filter outputs when using an example image.

5.2.3 Height estimation

When evaluating the results of the height estimation algorithm, it is easier to define quantitative performance metrics. One metric is to compare the altitude above the ellipsoid of each grid point in the resulting DEM, to the altitude of the corresponding point in the DEM from Lantmäteriet. A common error measurement in DEMs is Root Mean Square Error (RMSE) [32]. It is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{A}(i) - A(i)\right)^2},$$

where n is the number of grid points in the DEM, $A(i)$ denotes the altitude of the DEM from Lantmäteriet and $\hat{A}(i)$ the altitude of the estimated DEM at grid point i. The unit of the RMSE will be meters if A and $\hat{A}$ are specified in meters.

Another metric is to look at the proportion of points that have converged to their known correct position in northing, easting and altitude after height estimation is completed. A point is considered to have converged if the three-dimensional Euclidian distance from the correct location is less than 100 meters. Furthermore, the error in each of the mentioned coordinates can be evaluated as a function of the number of measurements supplied to the height estimation algorithm, giving convergence curves.

A subjective quality assessment can be made by visually inspecting the resulting DEM estimate and comparing it to the correct DEM.

5.2.4 Keypoint tracking

When tracking keypoints in real radar data, a few factors are considered for comparative reasons. The distribution of track lengths is considered, where a larger number of long tracks is preferable. Plotting the tracks in azimuth and range separately indicates how many erroneous tracks are present. Tracks that are close in either azimuth or range should be close to parallel. This separation will also indicate how noisy the measurements are.

Looking at the character of the found keypoints reveals if the estimated DEM will be true to reality or not. Peaks and valleys are important to detect as they will have large influence on the estimate. Finally, a subjective quality assessment is made from plotting the trajectories in the range-azimuth plane and comparing them with the radar data.

6 Implementation

All implementations are done in Matlab R2014b.

6.1 Geographical coordinate conversion

The following algorithms are used in succession for conversion between geographical coordinate systems. This has to be done in order to extract aircraft positions from the radar data, as well as transforming the estimated DEM into the same coordinate system as the evaluation DEM from Lantmäteriet. Both of the below algorithms are implemented so that whole lists of points can be converted in parallel, avoiding time-consuming loops.

6.1.1 Converting from SWEREF99 to geodetic coordinates

Longitude is found according to the simple calculation presented in 3.1.2.2. One iteration of the algorithm for finding latitude and altitude, described in the same section, using an ellipsoid with semimajor axis a and flattening constant f, is conducted as follows:

1. Compute the squared eccentricity $e^2 = f(2 - f)$.
2. Compute the following parameters:
   (a) $p = \sqrt{x^2 + y^2}$.
   (b) $e_c = \sqrt{1 - e^2}$.
   (c) $c = ae^2$.
   (d) $z_c = e_c|z|$.
3. Set initial values:
   (a) $S_0 = z$.
   (b) $C_0 = e_c p$.
4. Compute the following parameters:
   (a) $A_0 = \sqrt{S_0^2 + C_0^2}$.
   (b) $B_0 = \frac{3}{2} c S_0 C_0 \big((pS_0 - z_c C_0) A_0 - cS_o C_0 \big)$.
   (c) $S_1 = (z_c A_0^3 + cS_0^3) A_0^3 - B_0 S_0$.
   (d) $C_1 = (pA_0^3 - cC_0^3) A_0^3 - B_0 C_0$.
   (e) $C_c = e_c C_1$.
5. Compute latitude
   $\varphi = \arctan \frac{S_1}{C_c}$.

6. Compute altitude
   $h = \frac{pC_c + S_0 S_1 - a\sqrt{e_c^2 S_1^2 + C_c^2}}{\sqrt{S_1^2 + C_c^2}}$.

6.1.2 Converting from geodetic coordinates to SWEREF99TM

The SWEREF99TM projection, described in section 3.1.2.3, of a point at latitude φ and longitude λ onto an ellipsoid with semimajor axis a and flattening constant f, is done in the following way according to Lantmäteriet [16]:

1. Compute the squared eccentricity $e^2 = f(2 - f)$.
2. Compute the following parameters:
   (a) $n = \frac{f}{2-f}$.
   (b) $\hat{a} = \frac{a}{1+n}(1 + \frac{1}{4}n^2 + \frac{1}{64}n^4)$.
   (c) $A = e^2$.
   (d) $B = \frac{1}{6}(5e^4 - e^6)$.
   (e) $C = \frac{1}{120}(104e^6)$.
   (f) $D = \frac{1}{1260}1237e^8$.
   (g) $\beta_1 = \frac{1}{2}n - \frac{2}{3}n^2 + \frac{5}{16}n^3 + \frac{41}{180}n^4$.
   (h) $\beta_2 = \frac{13}{48}n^2 - \frac{3}{5}n^3 + \frac{577}{1440}n^4$.
   (i) $\beta_3 = \frac{61}{240}n^3 - \frac{103}{140}n^4$.
   (j) $\beta_4 = \frac{49561}{161280}n^4$.
3. Compute the conformal latitude
   $\varphi^* = \varphi - \sin\varphi \cdot \cos\varphi \cdot (A + B\sin^2\varphi + C\sin^4\varphi + D\sin^6\varphi)$.
4. Compute $\xi' = \arctan\left(\frac{\tan\varphi^*}{\cos(\lambda - 15)}\right)$.
5. Compute $\eta' = \operatorname{arctanh}\big(cos\varphi^* \cdot \sin(\lambda - 15)\big)$.
6. Compute northing
   $N = 0.996\hat{a}\big(\xi' + \beta_1 \sin 2\xi' \cosh 2\eta' + \beta_2 \sin 4\xi' \cosh 4\eta' + \beta_3 \sin 6\xi' \cosh 6\eta' + \beta_4 \sin 8\xi' \cosh 8\eta'\big)$.
7. Compute easting
   $E = 0.996\hat{a}\big(\eta' + \beta_1 \cos 2\xi' \sinh 2\eta' + \beta_2 \cos 4\xi' \sinh 4\eta' + \beta_3 \cos 6\xi' \sinh 6\eta' + \beta_4 \cos 8\xi' \sinh 8\eta'\big) + 500000$.

6.2 Radar data decoding

When decoding radar data, 8 bits, a so called bit word, is read at a time. There are synchronization words that specify what kind of information the following sequence

of words contain. For example aircraft positions, spoke azimuth or luminance data. Depending on which synchronization word is sent, the data is decoded in a specific manner according to classified documentation. The radar data files are too large to keep in system memory, so they have to be stored on the hard drive. A large drawback to this is that hard drive reading is a very slow process, as compared to the decoding of the data. This is circumvented by reading the data blockwise into memory. More specifically, the data is retrieved from the hard drive in blocks of 50 frames each. This means that every 50th frame will be significantly slower than the others, as can be seen in figure 20. However, the total execution time is significantly improved by doing this.

6.3 Scan conversion

After decoding a frame of radar data, the scan conversion is implemented as follows:

1. If the frame comes from a radar sweep from right to left, mirror it.
2. Convert all spoke origins from SWEREF99 to geodetic coordinates according to section 6.1.1.
3. Convert all spoke origins from geodetic coordinates to SWEREF99TM according to section 6.1.2.
4. Find azimuth from north for the first and last spoke in the frame.
5. Find maximum and minimum slant range in the frame.
6. Let the two azimuth values define two lines through the origin in the output image and calculate which pixels are between the lines.
7. Calculate which of these pixels that are between the minimum and maximum slant range from the origin.
8. Lookup the desired points in slant range and azimuth for all pixels according to the equations in section 3.2.2.
9. Make sure all angles are positive by adding 360 degrees to them.
10. Interpolate according to section 3.2.3 in order to find luminance values for all desired points.

6.4 Denoising

Both of the denoising methods described in section 3.3.2 are implemented. The implementations follow the descriptions in that section closely. However, one important aspect for fast execution is that the filtering is done by rearranging the image before filtering. The image is divided into distinct blocks, where each block is processed separately. Every block is supplied to the filter as a very wide matrix, where each column corresponds to the window around a specific pixel. The output should be

a row vector where each value specifies the desired value for the center pixel of the corresponding window. Filtering the image this way avoids having to loop through each pixel, which would otherwise have been the case, since the considered filters are nonlinear and therefore can not be written as convolutions.

6.5 Keypoint tracking

Keypoint tracking is carried out in the range-azimuth plane rather than in Cartesian coordinates. The reason for this is that it gives a rectangular image to work with as opposed to a circle sector. Furthermore, data resolution is lost during scan conversion since interpolations have to be performed, especially for long slant ranges. Since the data already has low resolution it is better to work with raw radar data.

6.5.1 Harris corner detection with digital image correlation

This version of keypoint tracking works by finding corners using Harris corner detection and then tracking them in subsequent frames by means of digital image correlation, as described in section 3.4.1.1. A check is made to see if the matches are ambiguous or not. If they are, the track is abandoned and a new one is initiated. The algorithm divides the frame into blocks and tries to keep the number of keypoints in each block as constant as possible. Because of this, the tracks are guaranteed to be spread out over the whole visible area, making sure that no area is ignored in the estimated DEM. If this was not the case, a subregion of the frame with high contrast could contain all tracks, so that they only contribute to the height estimation in that region.

For the first frame in a dataset, the algorithm works as follows:

1. Divide the frame into a specified number of blocks.
2. Use Harris corner detection to find a specified number of keypoints in each block.
3. Save the detected keypoint coordinates as the first elements in separate tracks.
4. Extract and save a descriptor of specified size for each track.
5. Save an indicator vector specifying that all tracks are active.

For all subsequent frames in the dataset:

1. For each active track:
   (a) Define a search area of predetermined size around the old keypoint location.
   (b) Calculate the C_{ZNSSD} defined in section 3.4.1.1.
   (c) Find the number of elements in the C_{ZNSSD} that are below a predetermined threshold.

(d) If there is only one point below the threshold, consider the match unambiguous:

i. Calculate the corresponding keypoint coordinates and save them to the corresponding track.

ii. Update the descriptor for this track.

(e) If there are more than one point below the threshold, or no points below the threshold, consider the match ambiguous:

i. Change the corresponding element of the indicator vector to specify the track as inactive.

ii. Divide the frame into the specified number of blocks.

iii. Count the number of active keypoints in each block.

iv. Find the block containing least active keypoints.

v. Use Harris corner detection to find a new keypoint in this block.

vi. Save the detected keypoint coordinates to a new track.

vii. Extract and save a descriptor of specified size for this track.

viii. Expand the indicator vector to specify that the new track is active.

(f) Continue to the next track.

2. Continue to the next frame.

6.5.2 Speeded-Up Robust Features

This keypoint tracking algorithm is based on the SURF algorithm, described in section 3.4.1.2. First, SURF is run for the whole frame in order to find good keypoints. Then it is run for the second frame to find keypoints. Then a matching process is done in order to find the continuation of each track in the region close to it. If no sufficiently good match is found, a new track is started nearby.

For the first frame in a dataset:

1. Use SURF to find a specified number of keypoints.
2. Save the detected keypoint coordinates as the first elements in separate tracks.
3. Save the 64-element SURF descriptor of each track.
4. Save an indicator vector specifying that all tracks are active.

For all subsequent frames in the dataset:

1. Use SURF to find many more keypoints than there are active tracks.
2. For each active track:

(a) Select all new keypoints that are within a specified distance of the old one in the range-azimuth plane.

(b) Calculate the Euclidian distance from the track descriptor to the descriptor of each of the selected keypoints.

(c) If the shortest distance is below a specified threshold, consider this keypoint a match:

i. Save the new keypoint coordinates as the next element in the track.

ii. Update the 64-element SURF descriptor of this track.

(d) If the shortest distance is not below a specified threshold, consider this track lost:

i. Change the corresponding element of the indicator vector to specify the track as inactive.

ii. Pick the closest new keypoint in the azimuth-range plane.

iii. Save the keypoint coordinates as the first elements in a new track.

iv. Save the 64-element SURF descriptor of this track.

v. Expand the indicator vector to specify that the new track is active.

(e) Continue to the next track.

3. Continue to the next frame.

6.6 Height estimation

The height estimation algorithm described in 3.4.2 is implemented by Zoran Sjanic at Saab. It is basically a Recursive Least Squares (RLS) algorithm for finding the best intersection point for all circles defined by each track.

It works in a Cartesian coordinate system, where the origin is defined as the point on the ellipsoid that is straight down from the aircraft position, in the first frame of the dataset. The x-axis points along the aircraft's forward direction in the horizontal plane. The y-axis points to the left from the aircraft direction in the horizontal plane, and the z-direction points up. For example, a point at 5000 meters altitude, that is 100 meters ahead of the aircraft position, would be $(100, 0, 5000)$. There is no rotation of the coordinate axes as the aircraft turns, so coordinates in subsequent frames have to be rotated back into this system. All calculations use the assumption that all keypoints are so close that the earth surface can be approximated as a plane, greatly simplifying calculations.

Inputs to the algorithm are keypoint tracks containing azimuth and slant range measurements, along with the aircraft positions from which the keypoints have been detected. The output will be an estimated position in the aforementioned Cartesian coordinate system for each keypoint.

Parameters to the algorithm are guesses of the initial insecurities (standard deviations) of keypoint positions along each of the Cartesian axes. These will be updated for each new iteration of the RLS algorithm. After some initial trial and error, it was concluded that a standard deviation of 300 meters along each axis was suitable.

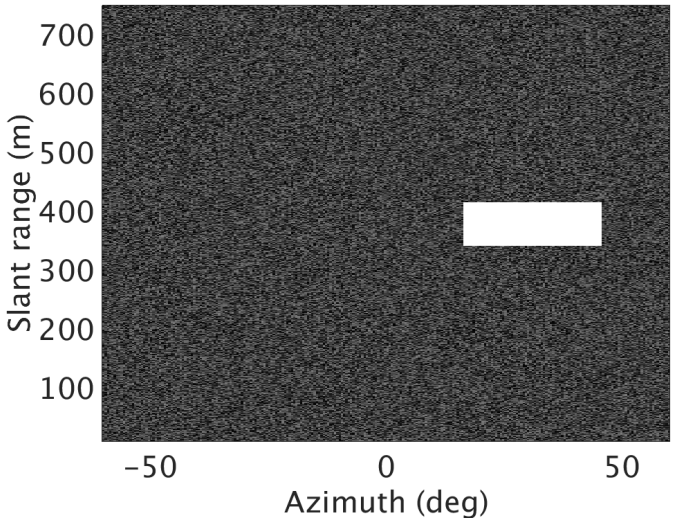
After each keypoint position is estimated, a bilinear interpolation is done to evaluate the DEM at each grid point. The reason for not using bicubic interpolation is that this might give the illusion of having much knowledge in an area, just because the DEM looks smooth there. With bilinear interpolation, it will be easier to see that a region is lacking keypoints, since there will be a large flat surface in the DEM.

7 Experiments and results

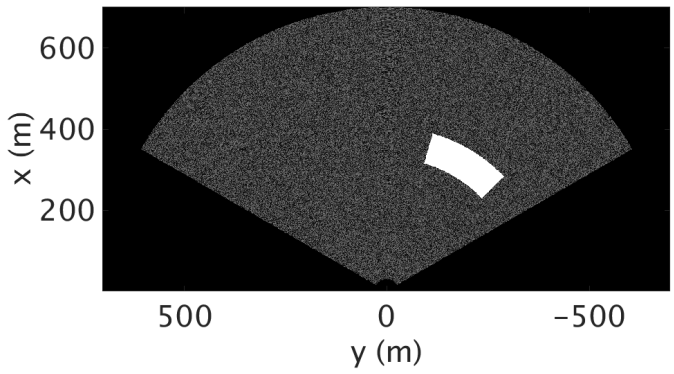
7.1 Scan conversion

7.1.1 Validation using test image

Trying the scan conversion on a simple synthetic test image gives the results in figure 19. The algorithm seems to work as intended. When looking at actual radar data from the Gotland dataset the results are also reasonable, so the scan conversion seems to work even for real data.



(a)



(b)

Figure 19: (a) A synthetic test frame in polar coordinates. (b) Output from the scan conversion algorithm.

7.1.2 Comparison of interpolation schemes

Comparing scan conversion results using both bilinear and bicubic interpolation gives visually equivalent results when working on real radar data. Unfortunately, these results can not be published in this report. However, a comparison of the execution times for the scan conversion when using the different interpolation schemes is shown in figure 20.

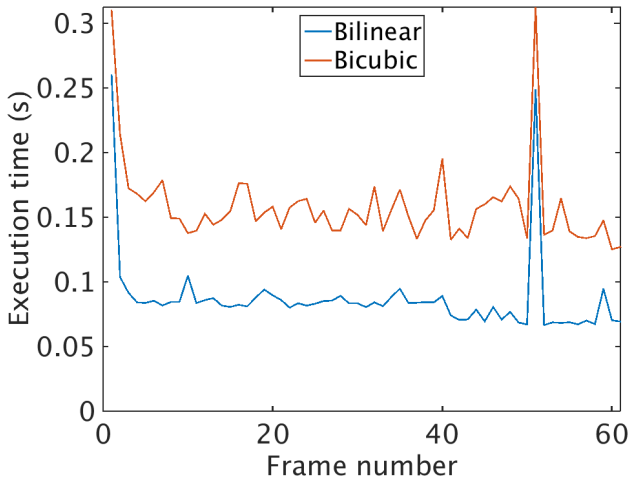
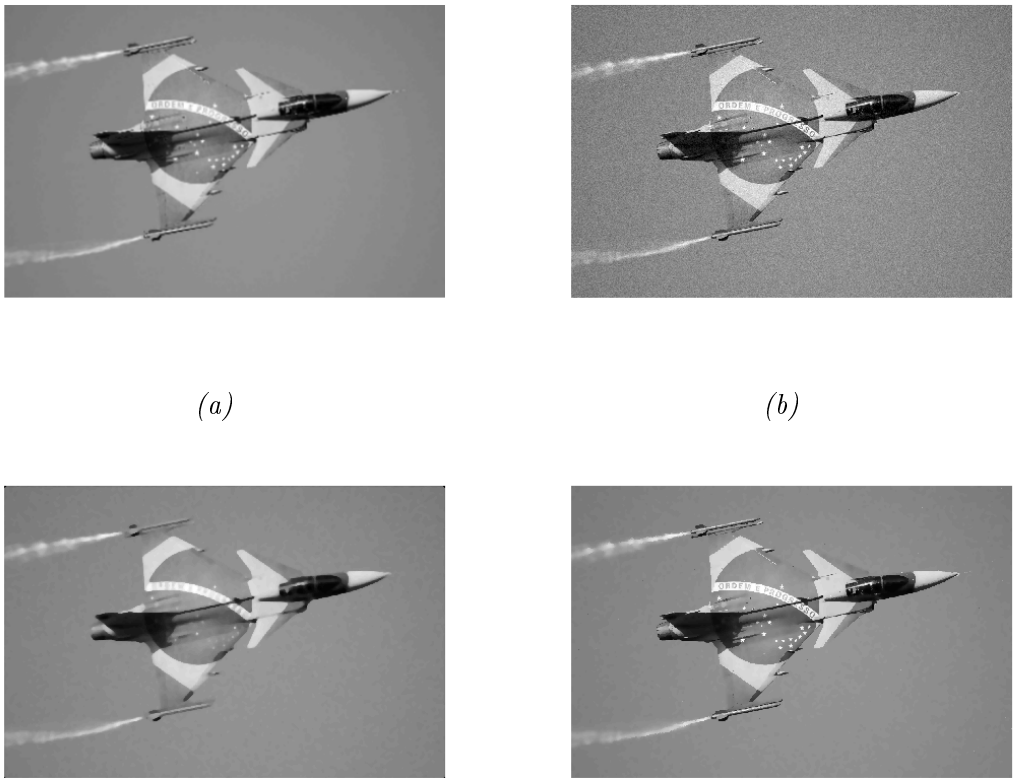


Figure 20: Execution time for scan conversion frame by frame when using bilinear and bicubic interpolation.

7.2 Denoising

To test the denoising filters an image of a Gripen painted with the Brazilian flag, with added Gaussian noise, is used as input. The reason for this is that real radar data can not be included in this report. Number of iterations, filter size and σ parameter for the sigma filter were tuned until as much noise as possible was suppressed without losing edge information in the image. Then the median filter parameters were tuned until a similar reduction in noise was achieved. The resulting parameters were a filter size of 7×7 pixels, three iterations and a σ parameter of 10. Filter outputs for these parameters are shown in figure 21. The sigma filter is clearly superior at preserving details, note specially that the stars in the Brazilian flag are preserved, as well as the text. The contrast around the edges in the image is even increased in comparison to the original image.

After tuning the filter parameters by trial and error using the keypoint tracking algorithms, the following parameters were chosen: A filter size of 7×7 pixels, one iteration and a σ parameter of 10. The biased sigma filter was deemed unnecessary for the available radar data.



(a) *(b)*

(c) *(d)*

Figure 21: Results from the two considered denoising filters. (a) The original image. (b) The image with added Gaussian noise. (c) Output from the median filter. Edges are well-preserved but fine details such as stars and text have been lost. (d) Output from the sigma filter. Fine details such as stars and text have been preserved and the edges have been sharpened.

7.3 Height estimation

Below, a number of experiments for validating and testing the height estimation algorithm, according to the outline in section 5.1, are presented,. The results are evaluated according to the performance metrics given in section 5.2.3.

7.3.1 Synthetic data without noise

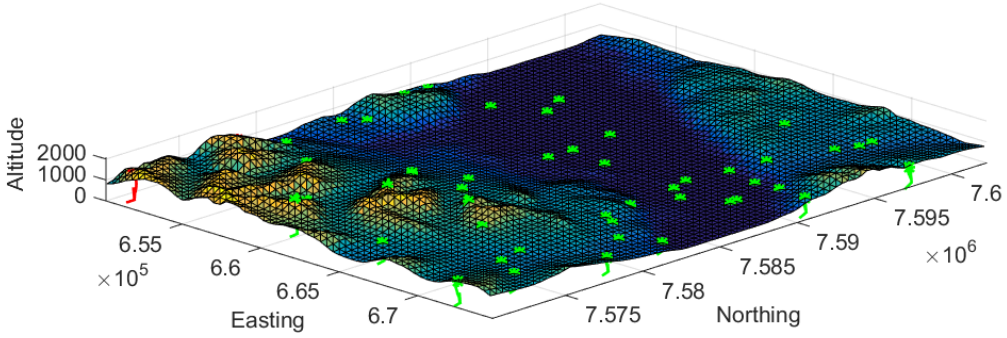
The following experiments are carried out on simulated data without added noise. Parameters are varied one at a time in order to evaluate their influence on the estimation.

7.3.1.1 Varying number of tracks

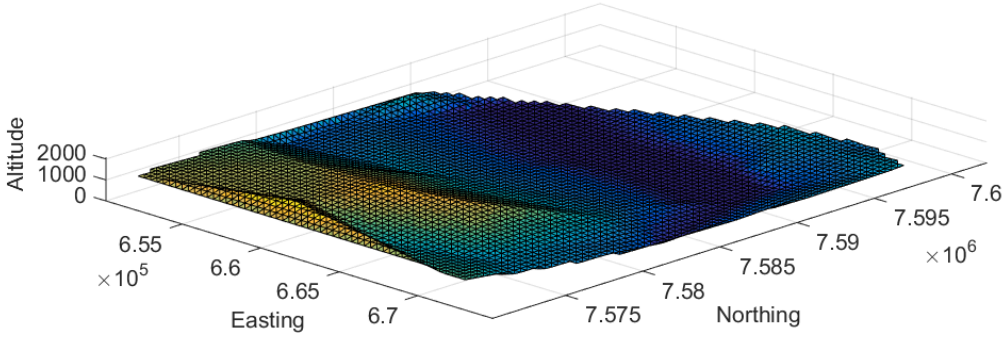
Initially, a grid consisting of a specified number of keypoints is selected randomly in the Abisko DEM. A straight aircraft trajectory with 75 frames of data at 7000 meters altitude is simulated. This experiment is run for 50, 100, 250 and 500 keypoints to investigate how the number of tracks affect the resulting height estimates. In figures 22 to 25, the original DEMs along with their corresponding estimates are shown. In the original DEMs the trajectories of the estimated keypoint positions are shown. Green means that the final estimate for that keypoint is within 100 meters of its correct position and red means it is not. The final RMSE and point convergence statistics are presented in table 1. A higher number of keypoint tracks leads to a better DEM estimate, just as expected.

Keypoints	**RMSE**	**Points converged**
50	148 m	96%
100	121 m	95%
250	101 m	94.8%
500	79 m	96%

Table 1: Numerical results from estimating the Abisko DEM using tracks of length 75 and varying the number of keypoints.

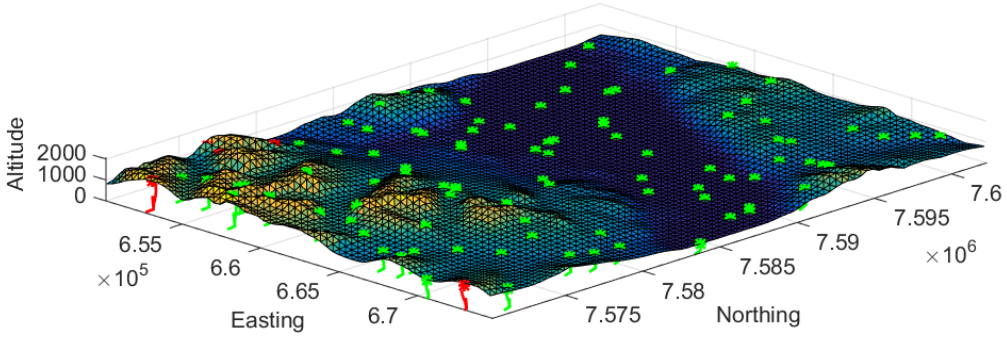


(a)

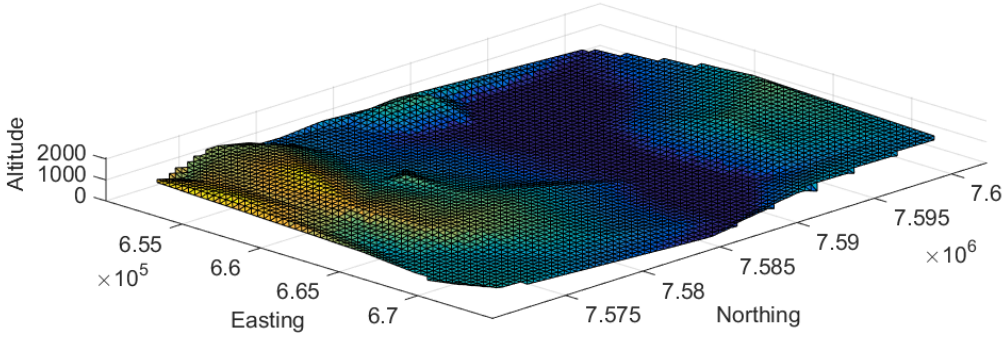


(b)

Figure 22: DEM estimation using 50 keypoint tracks and a straight aircraft trajectory. (a) Original DEM with estimated keypoint positions. (b) Estimated DEM.

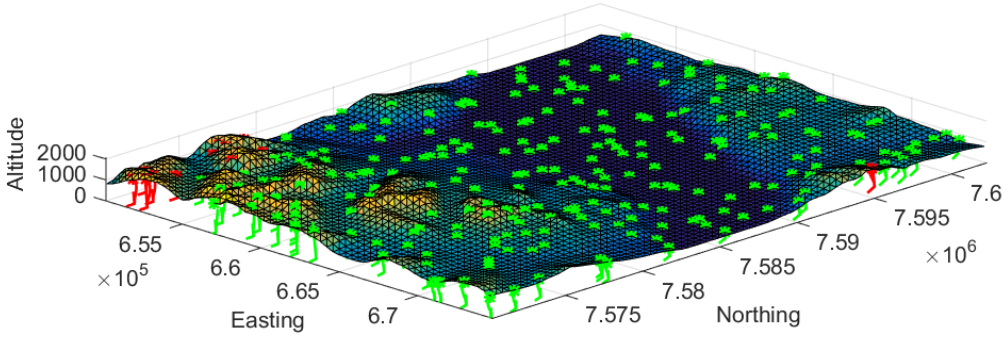


(a)

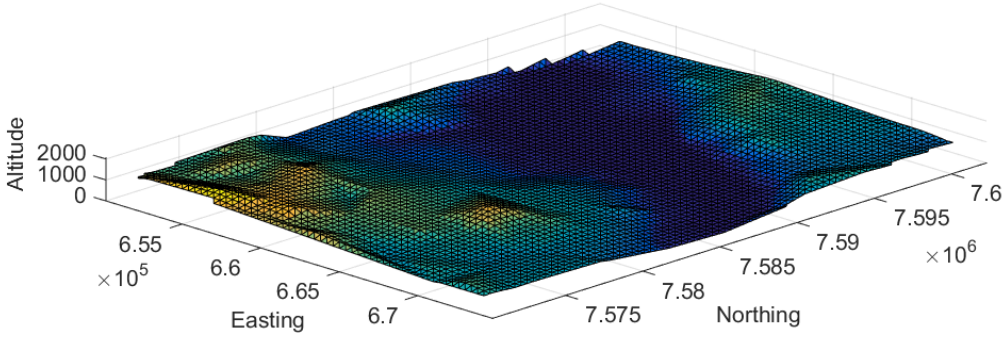


(b)

Figure 23: DEM estimation using 100 keypoint tracks and a straight aircraft trajectory. (a) Original DEM with estimated keypoint positions. (b) Estimated DEM.

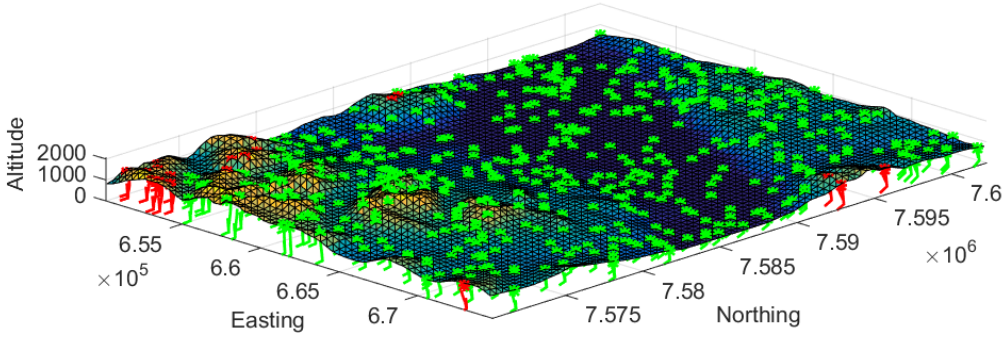


(a)

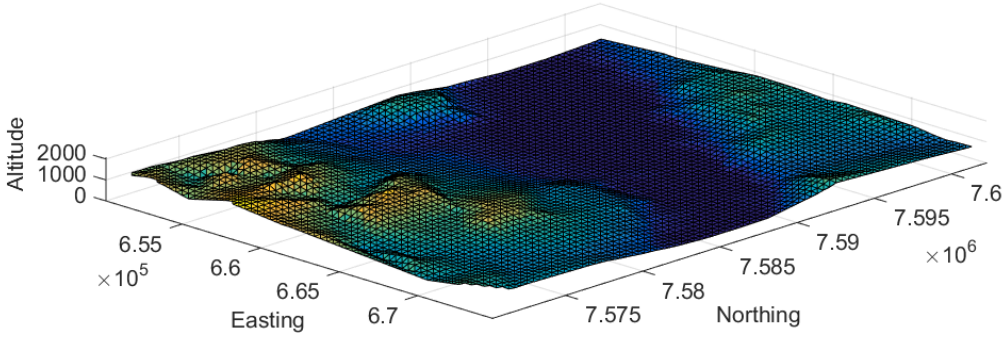


(b)

Figure 24: DEM estimation using 250 keypoint tracks and a straight aircraft trajectory. (a) Original DEM with estimated keypoint positions. (b) Estimated DEM.



(a)



(b)

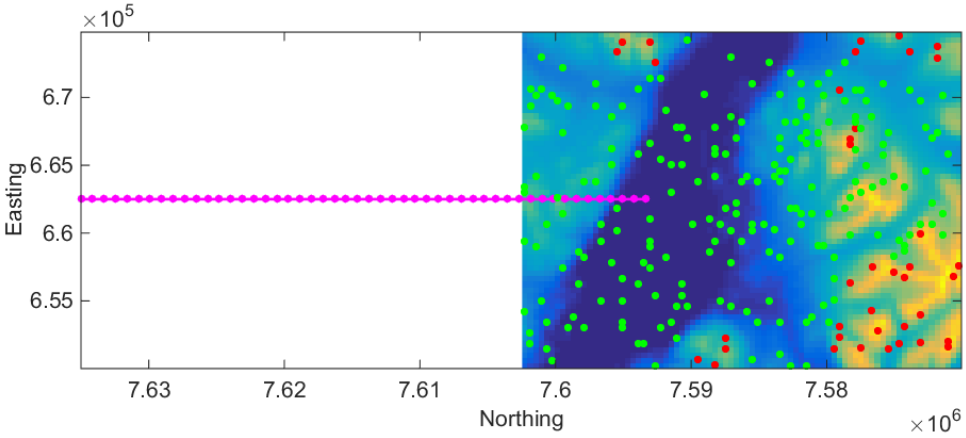
Figure 25: DEM estimation using 500 keypoint tracks and a straight aircraft trajectory. (a) Original DEM with estimated keypoint positions. (b) Estimated DEM.

7.3.1.2 Varying aircraft trajectory

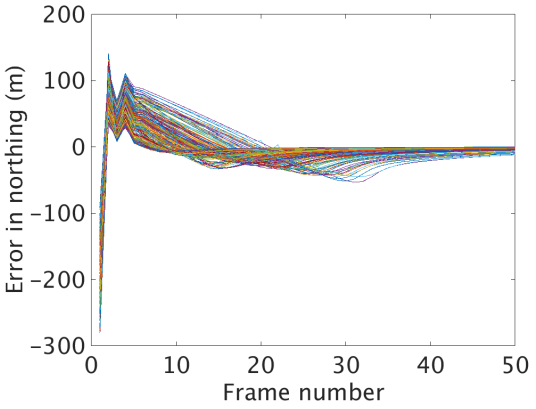
Next, the number of keypoints is fixed to 250 and the aircraft trajectory is varied instead. Three different shapes are considered: one straight, one where the aircraft turns slightly left and one where it turns slightly right. All of these trajectories are simulated at 5000 meters altitude. Since this experiment aims at comparing convergence properties, the same keypoints are selected every time. The results of these three experiments are presented in figures 26 to 28. In each figure, a top down view of the area is shown, along with all selected keypoints. As before, green means that it is within 100 meters of its correct position, and red means it is not. Numerical results are presented in table 2. Keypoint convergence seems better when using a curved aircraft trajectory.

Trajectory	**RMSE**	**Points converged**
Straight	118 m	84.4%
Sligthly left	107 m	88.8%
Slightly right	139 m	86.8%

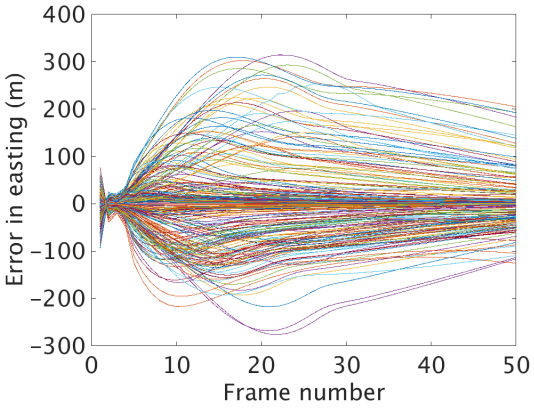
Table 2: Numerical results from trying different aircraft trajectories with a fixed number of 250 keypoints.



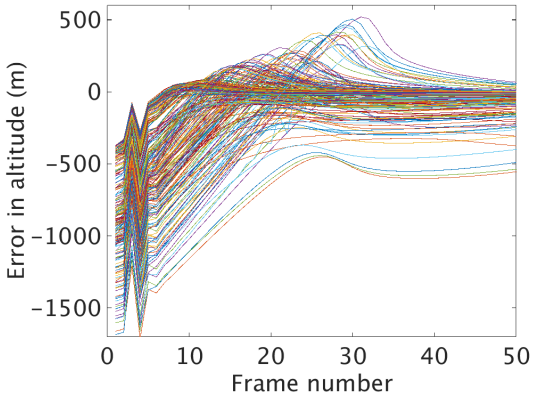
(a)



(b)

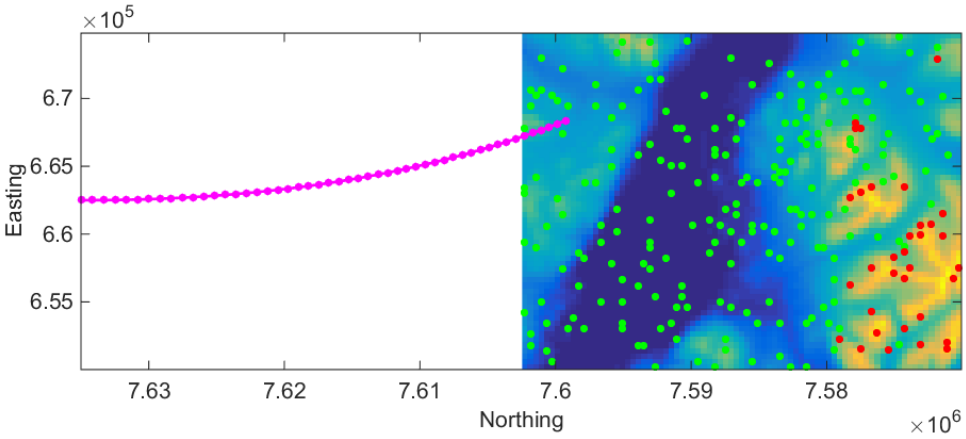


(c)

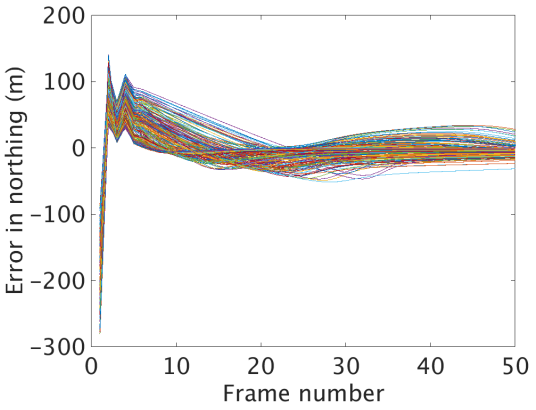


(d)

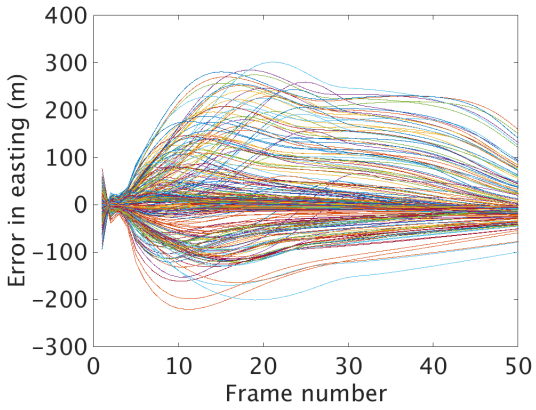
Figure 26: Convergence properties when flying in the straight line drawn in purple. (a) Top down view, green points have converged and red points have not. (b) Convergence in northing. (c) Convergence in easting. (d) Convergence in altitude.



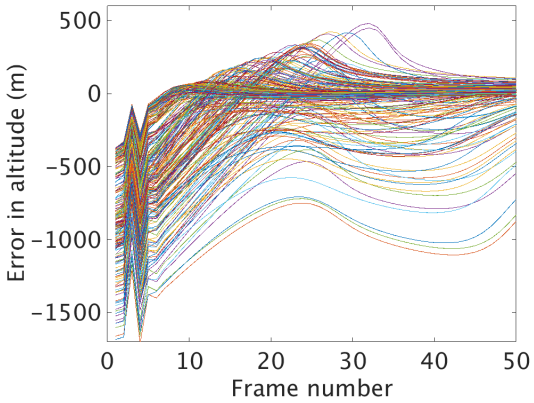
(a)



(b)

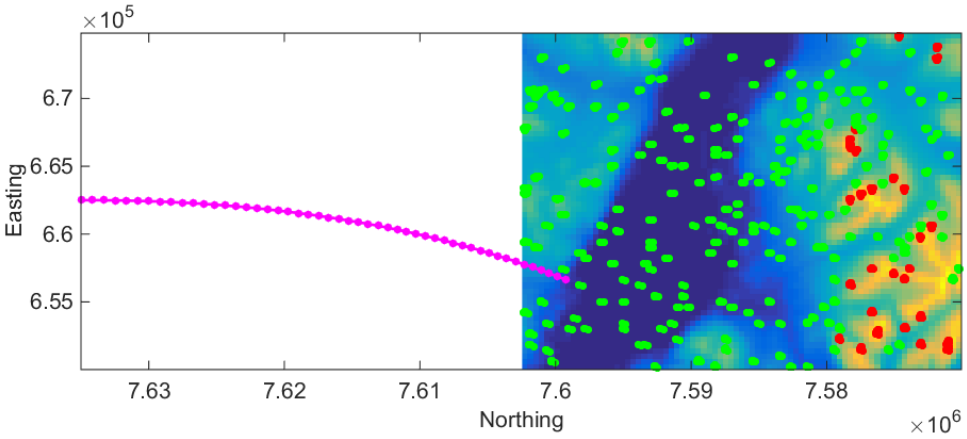


(c)

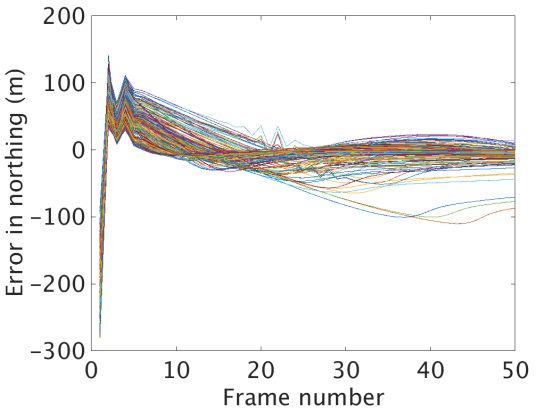


(d)

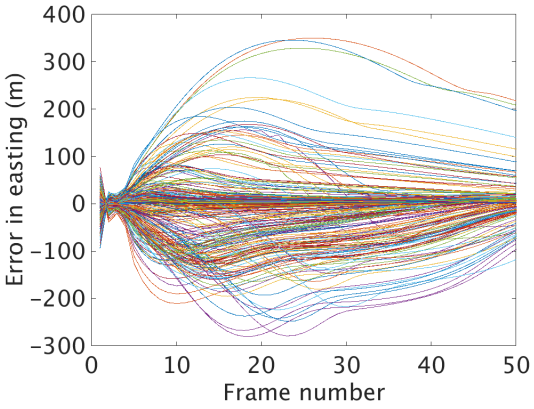
Figure 27: Convergence properties when using the leftward trajectory in purple. (a) Top down view, green points have converged and red points have not. (b) Convergence in northing. (c) Convergence in easting. (d) Convergence in altitude.



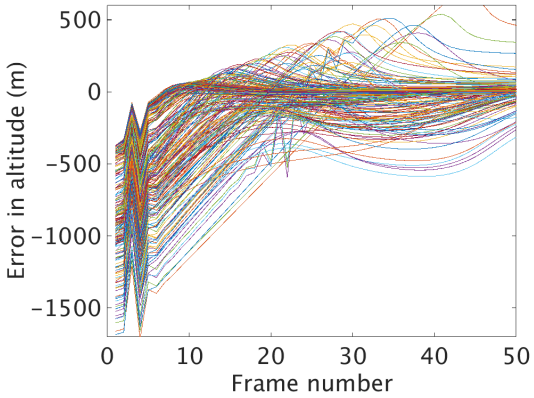
(a)



(b)



(c)



(d)

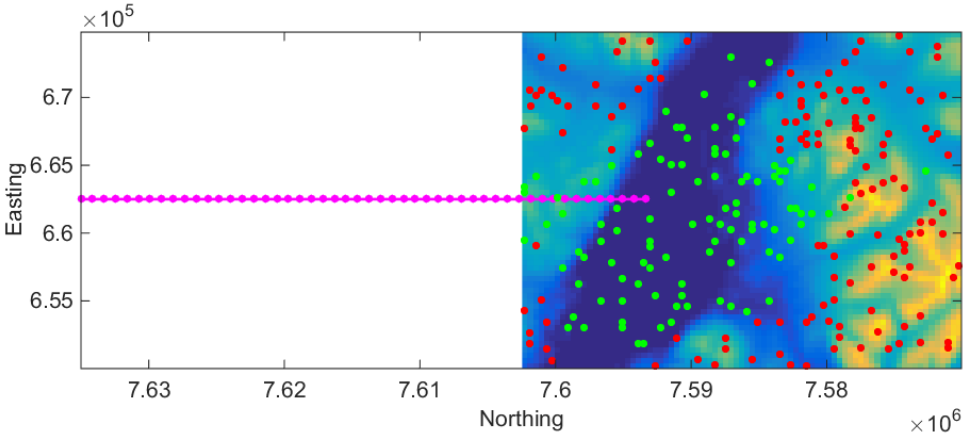
Figure 28: Convergence properties when using the rightward trajectory in purple. (a) Top down view, green points have converged and red points have not. (b) Convergence in northing. (c) Convergence in easting. (d) Convergence in altitude.

7.3.1.3 Varying altitude

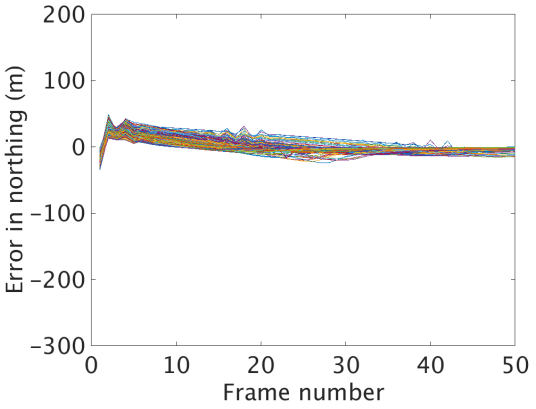
When instead varying the altitude and using a fixed number of 250 keypoints, the results presented in figures 29 to 31 are obtained. Numerical results are shown in table 3. As before, the selected keypoints are the same in each experiment for comparability. Interestingly, the convergence properties seem worse when flying at lower altitudes.

Altitude	**RMSE**	**Points converged**
10000 m	89 m	98%
6000 m	102 m	87.6%
2000 m	285 m	42.4%

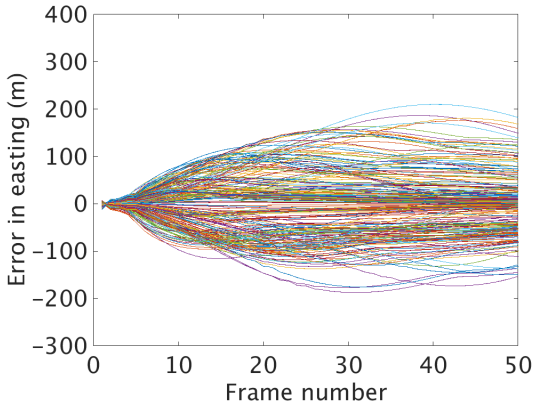
Table 3: Numerical results from trying different aircraft altitudes with a fixed number of 250 keypoints.



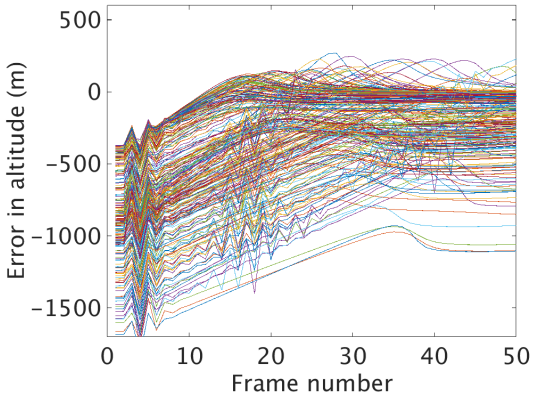
(a)



(b)

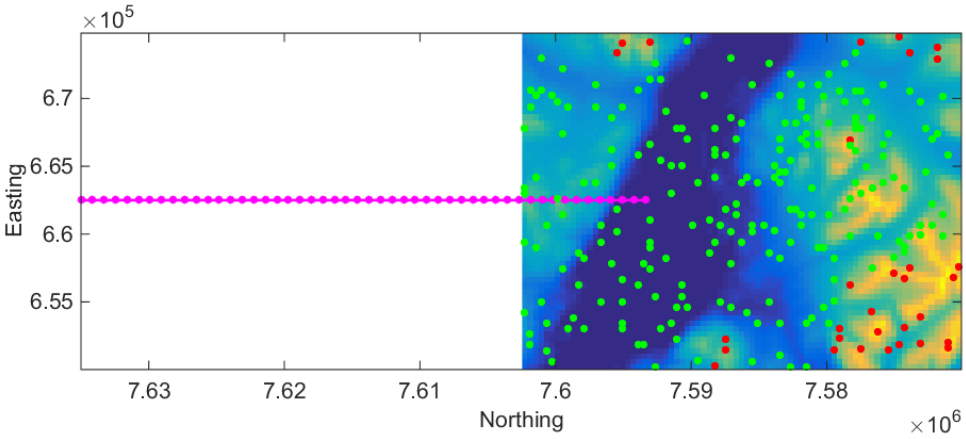


(c)

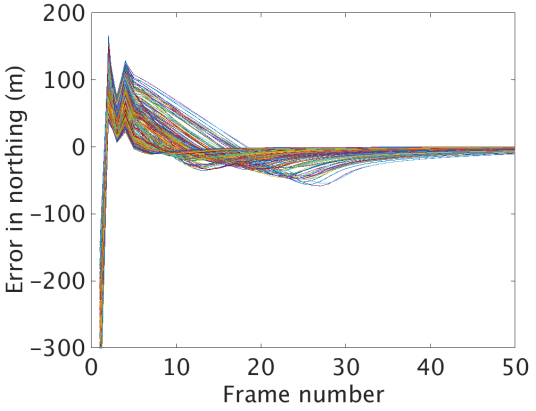


(d)

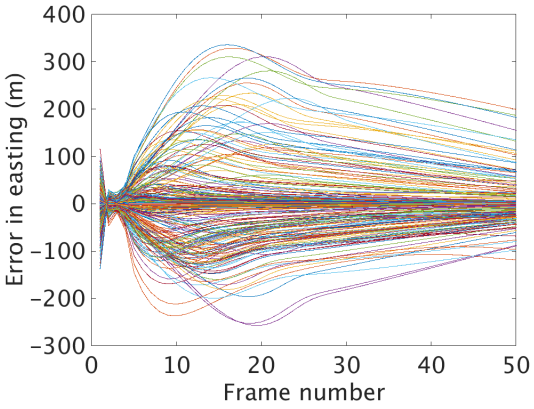
Figure 29: Convergence properties when flying straight at 2000 meters altitude. (a) Top down view, green points have converged and red points have not. (b) Convergence in northing. (c) Convergence in easting. (d) Convergence in altitude.



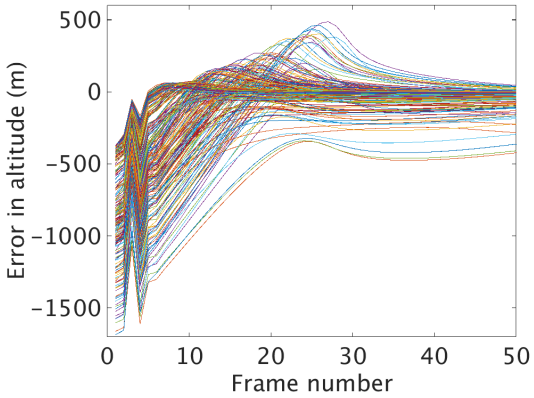
(a)



(b)

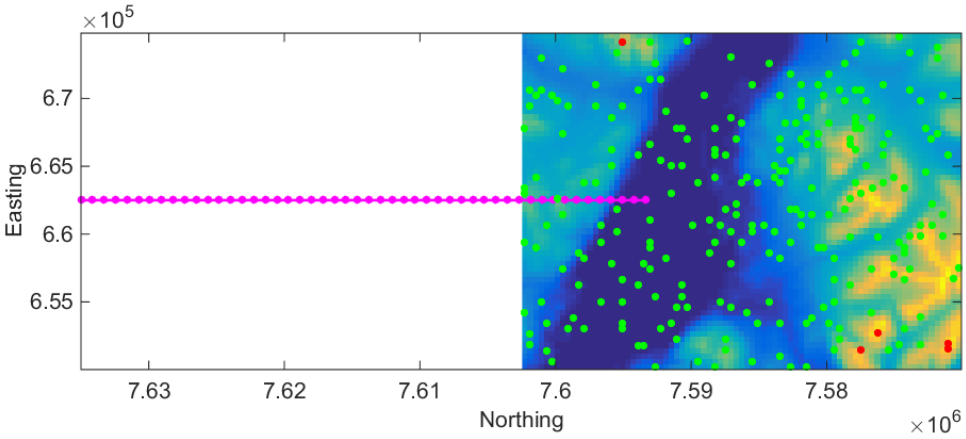


(c)

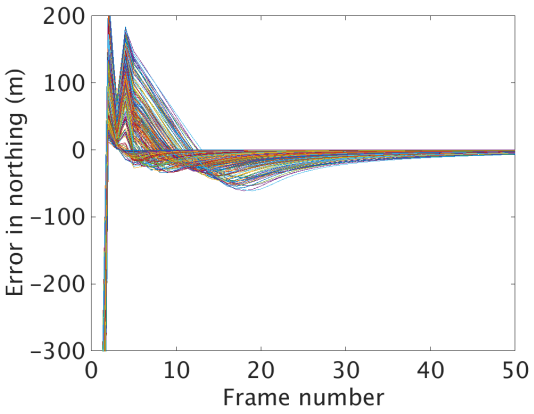


(d)

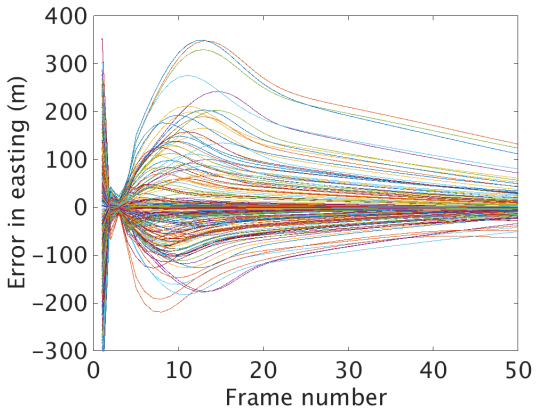
Figure 30: Convergence properties when flying straight at 6000 meters altitude. (a) Top down view, green points have converged and red points have not. (b) Convergence in northing. (c) Convergence in easting. (d) Convergence in altitude.



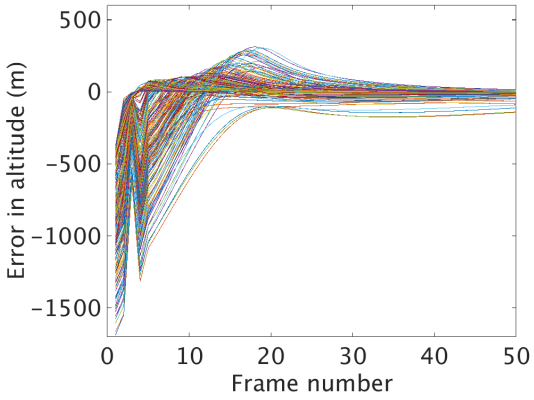
(a)



(b)



(c)



(d)

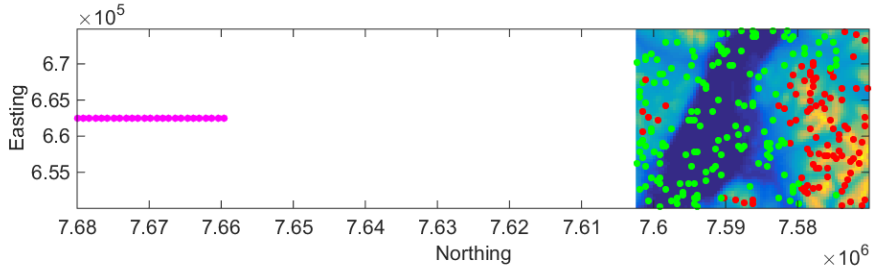
Figure 31: Convergence properties when flying straight at 10000 meters altitude. (a) Top down view, green points have converged and red points have not. (b) Convergence in northing. (c) Convergence in easting. (d) Convergence in altitude.

7.3.1.4 Varying track length

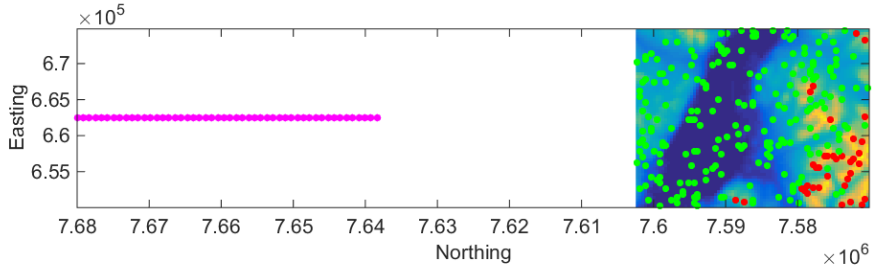
Next, varying the track length gives the results shown in table 4 and figure 32. The tested track lengths are 25, 50, 75 and 100 frames. It is clearly visible that more points have time to converge if the corresponding tracks are longer.

Track length	RMSE	Points converged
25	146 m	62.4%
50	103 m	85.2%
75	99 m	93.2%
100	96 m	100%

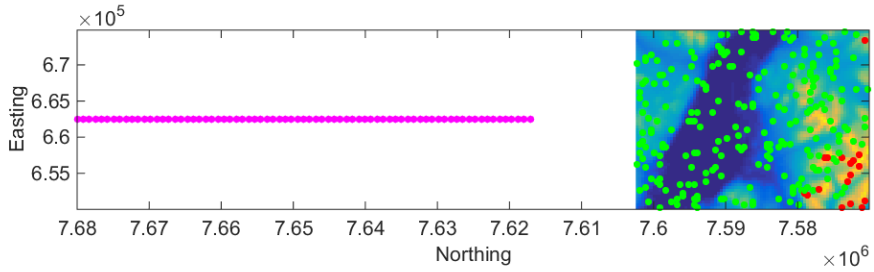
Table 4: Numerical results from trying different track lengths with a fixed number of 250 keypoints.



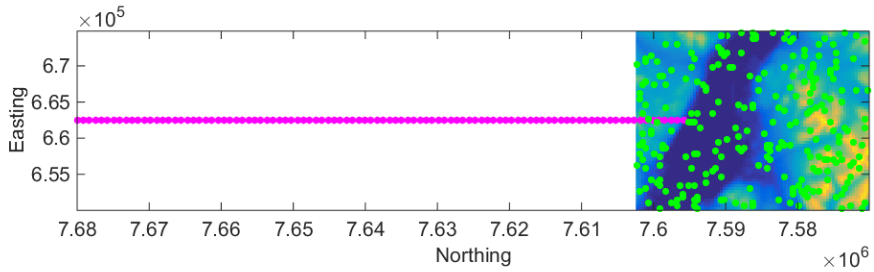
(a)



(b)



(c)



(d)

Figure 32: Results from varying the track length. (a) 25 frames. (b) 50 frames. (c) 75 frames. (d) 100 frames.

7.3.2 Synthetic data with noise

Using a straight aircraft trajectory and 500 keypoint tracks, the algorithm is evaluated with added measurement noise, simulating different data resolutions. The same procedure as above is performed, but with the addition of Gaussian noise both in the slant range measurements and azimuth angles. Noise power is varied by changing the standard deviation of the underlying Gaussian distribution. The considered standard deviations in slant range are 25, 50 and 100 meters. In azimuth angle the corresponding values are 0.25, 0.5, 0.75 and 1 degrees. Results from fixing the standard deviation in slant range and varying the standard deviation in azimuth can be seen in figure 33. It is clearly visible that in order to get high convergence proportion for short tracks, noise must be low for both measurements.

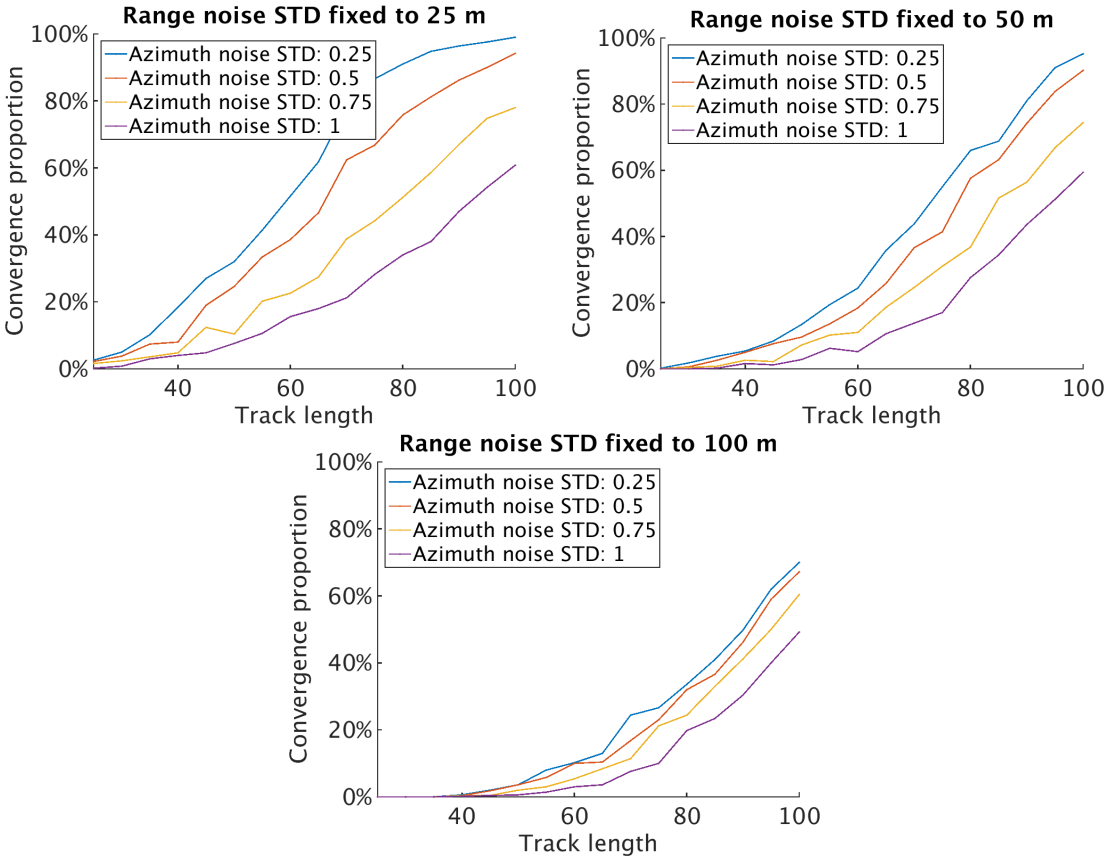


Figure 33: Curves showing how the proportion of converged points vary with track lengths when fixing range noise levels and varying azimuth noise levels.

7.4 Keypoint tracking

In this section a few experiments for validating and testing the two keypoint tracking algorithms, according to the outline in section 5.1, are presented. The results are evaluated according to the performance metrics given in section 5.2.4. Figures containing all tracks are shown as well as figures only showing those tracks that are

longer than five frames. In order to not reveal the range and resolution of the radar, the figures have been censored and the slant range scales are omitted.

7.4.1 Harris corner detection with digital image correlation

7.4.1.1 Östergötland dataset

When keeping the number of keypoints in each frame to 100, the tracks shown in figure 34 are obtained. Most tracks seem reasonable, but not very long in general. The detected keypoints are mostly lakes, coastline and edges of urban areas, but some seem to be other terrain features, most likely hills. However, the longest tracks are lakes and coastline. The white area in the figure is present since the pilot realigns the radar to look closer to the aircraft, which means the radar data no longer covers the top part. Plots of these tracks separated into slant range and azimuth are presented in figure 35. Here we see that most tracks are parallel in slant range, which is expected. In azimuth they are spreading out from the forward direction, which is also reasonable, since the angle to a point that is not directly in front of the aircraft will increase as it comes closer. The sudden break in all tracks is also due to the pilot realigning the radar. Histograms of the lengths of the detected tracks are seen in figure 36. We see that most tracks that are longer than five frames are between 5 and 20 frames long, with very few exceptions.

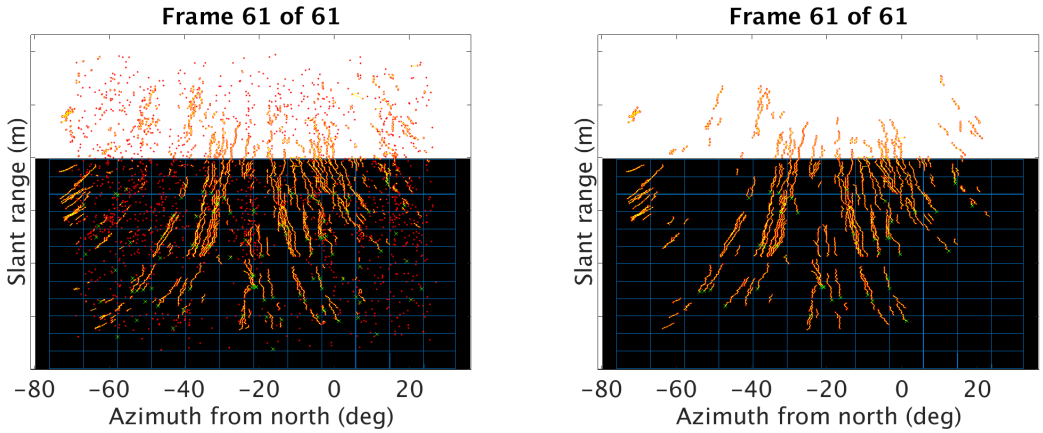


Figure 34: Tracks found in the Östergötland dataset by the Harris keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

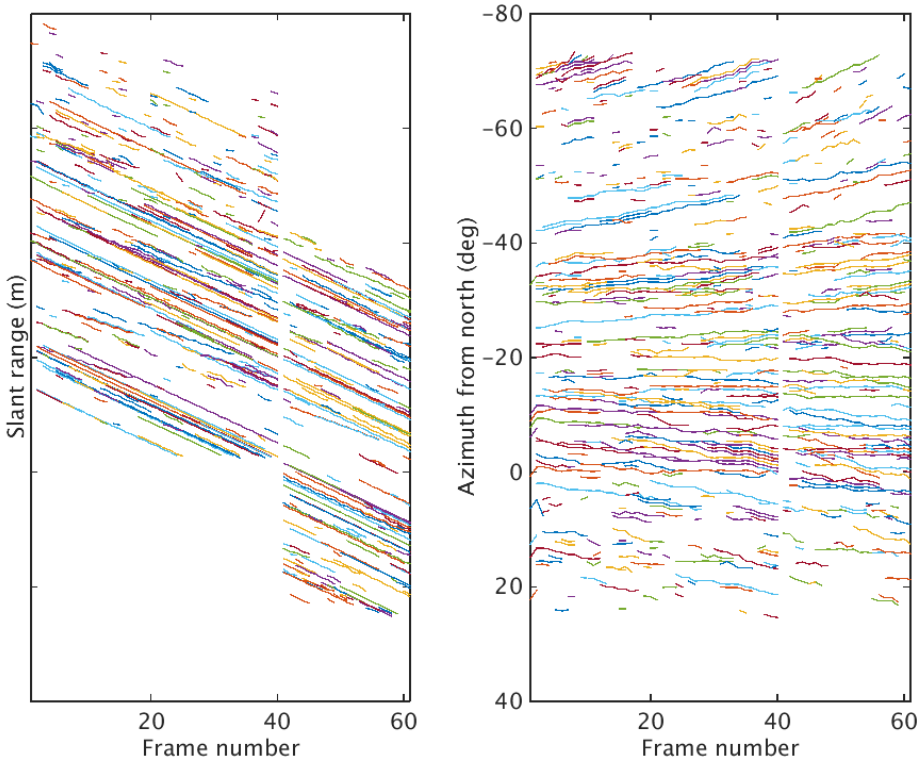
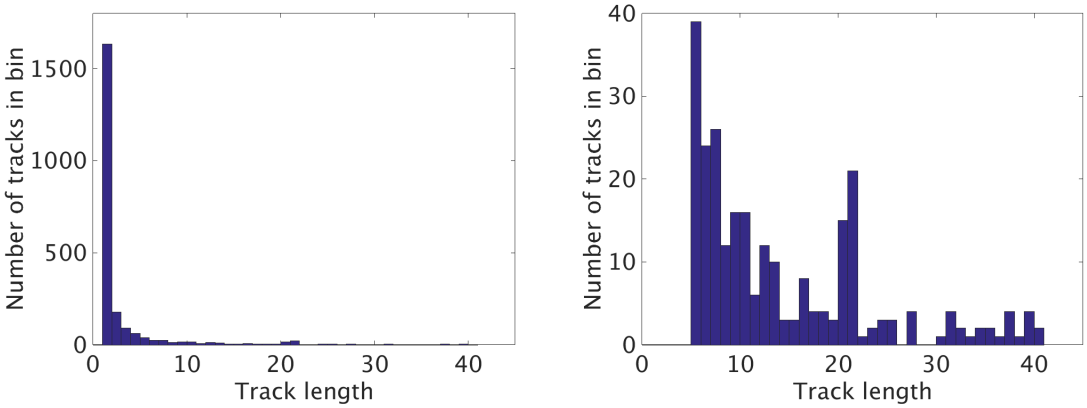


Figure 35: Tracks found in the Östergötland dataset by the Harris keypoint tracking algorithm, separated into slant range and azimuth tracks.



(a) *(b)*

Figure 36: Histograms of the lengths of the found tracks in the Östergötland dataset by the Harris keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

7.4.1.2 Stockholm dataset

When keeping the number of keypoints in each frame to 100 the tracks shown in figure 37 are obtained. In the Stockholm area the terrain is more varying than in Östergötland, which shows up in a better spread of the tracks over land, in the left side of the frame. The detected keypoints are mostly lakes, coastline and edges of urban areas, but some also seem to be hills. However, the longest tracks are lakes and coastline. Tracks in the right part of the frame are ships out at sea, and the white area in the figure is present since the airplane turns slightly right in the end of the dataset. Plots of the tracks separated into slant range and azimuth are presented in figure 38. Most tracks are parallel in slant range and spreading in azimuth, but there are some clearly erroneous ones. These are caused by the heavy sea clutter present in this dataset. Histograms of the lengths of the detected tracks are seen in figure 39. The track length distribution is similar to that in the previous dataset.

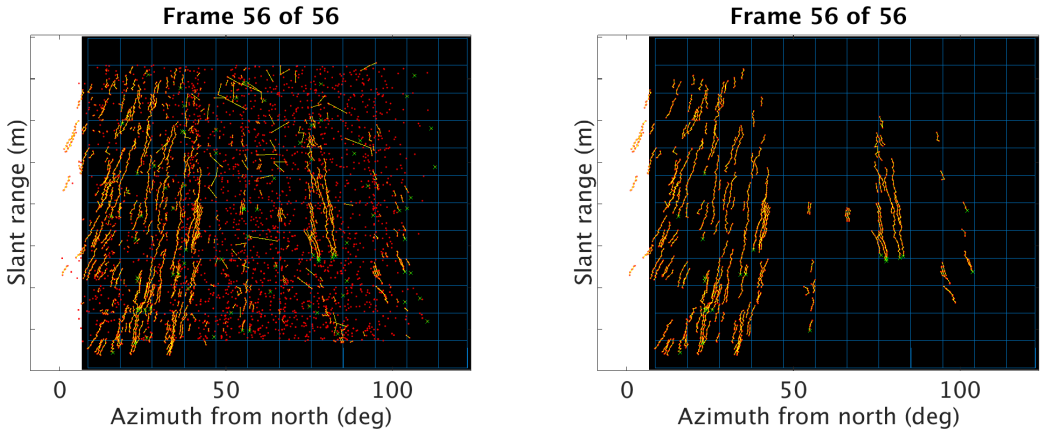


Figure 37: Tracks found in the Stockholm dataset by the Harris keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

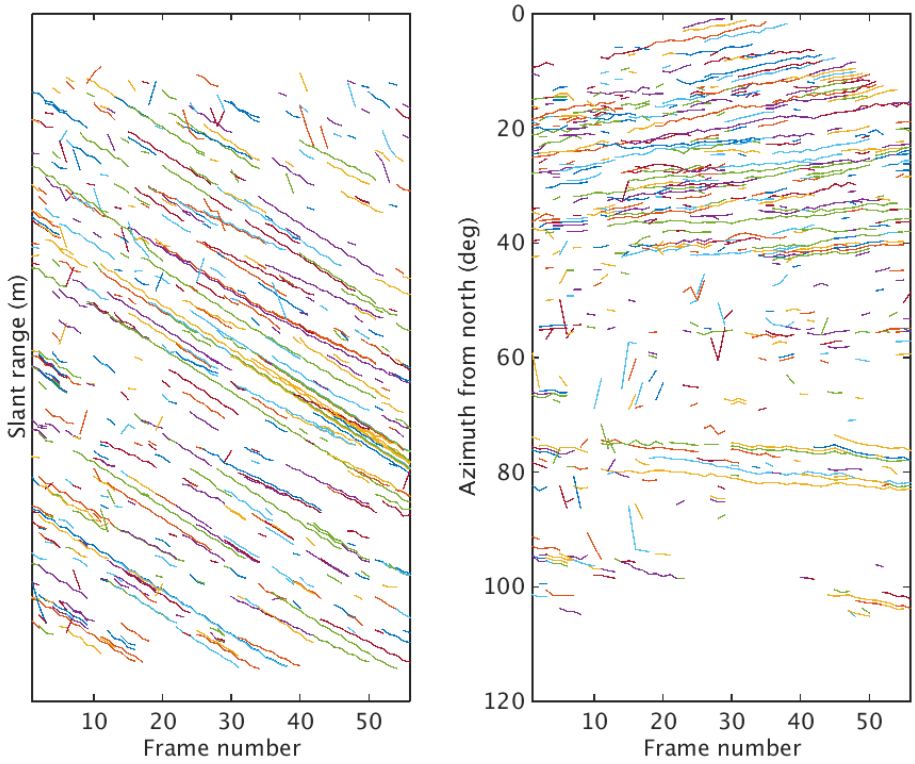
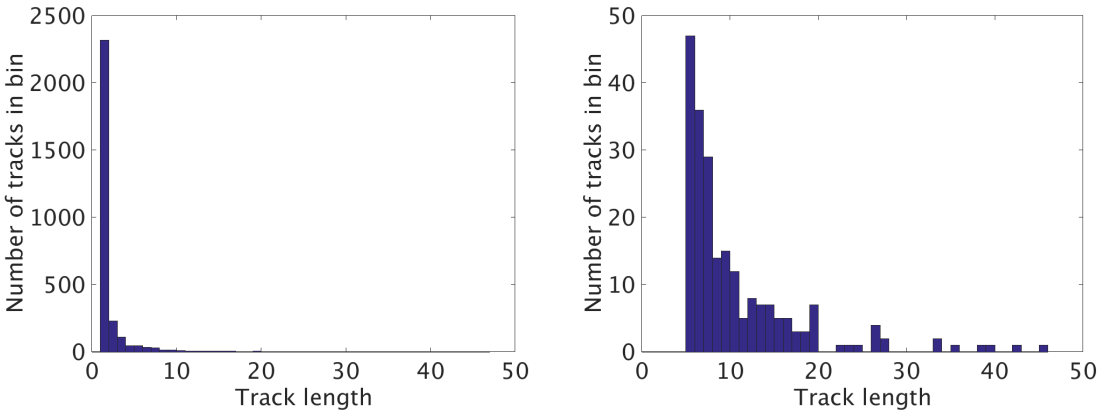


Figure 38: Tracks found in the Stockholm dataset by the Harris keypoint tracking algorithm, separated into slant range and azimuth tracks.

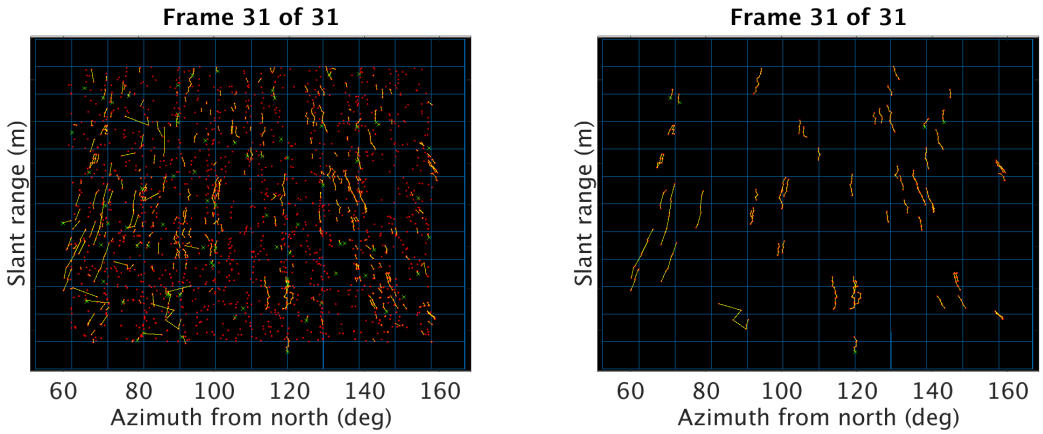


(a) *(b)*

Figure 39: Histograms of the lengths of the found tracks in the Stockholm dataset by the Harris keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

7.4.1.3 Gotland dataset

Once more keeping the number of keypoints in each frame to 100, the tracks shown in figure 40 are obtained. Gotland is very flat, which leads to quite bad tracks. The tracks that seem reasonable are following coastline, the city of Visby and boats out at sea. Plots of the tracks separated into slant range and azimuth are presented in figure 41. Most tracks are parallel in slant range and spreading in azimuth, but there are some clearly erroneous ones. Histograms of the lengths of the detected tracks are seen in figure 42. Very few tracks of significant length are obtained, with most being just above five frames long.



(a) *(b)*

Figure 40: Tracks found in the Gotland dataset by the Harris keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

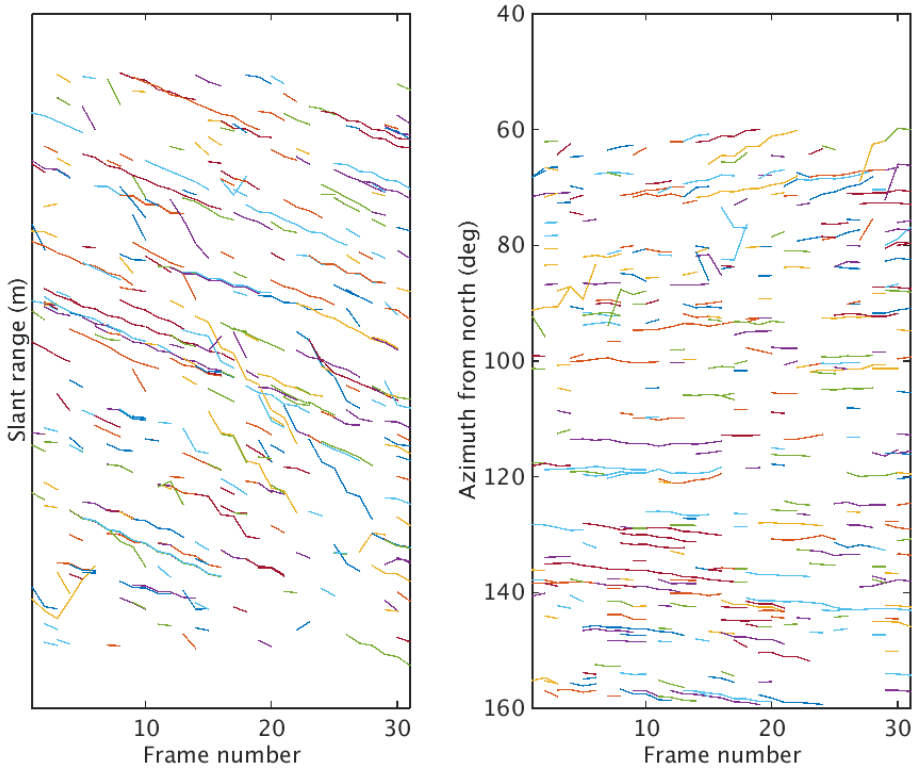
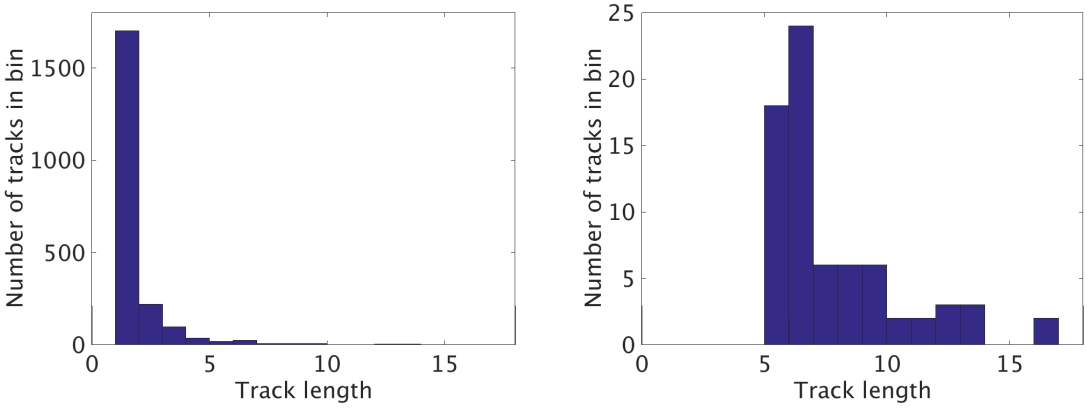


Figure 41: Tracks found in the Gotland dataset by the Harris keypoint tracking algorithm, separated into slant range and azimuth tracks.



(a) *(b)*

Figure 42: Histograms of the lengths of the found tracks in the Gotland dataset by the Harris keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

7.4.2 Speeded-Up Robust Features

Trying this algorithm on the Östergötland dataset and keeping the number of keypoints in each frame to 100 gives the tracks presented in figure 43. The tracks are shown separated into slant range and azimuth in figure 44. Generally, the tracks are long, but because of the low data resolution the keypoints are detected at large scale, which leads to bad accuracy. The reason why there are no tracks in the lower part of the frame is that the pilot has realigned the radar to focus on the ground closer to the aircraft; the algorithm does not yet have any means of placing new tracks in the newly revealed area. This realignment is also the reason why there is a white area at the top of the figure. Looking at the track length histograms in figure 45, it is clear that this algorithm is capable of tracking points for a longer time, as compared to the previous algorithm.

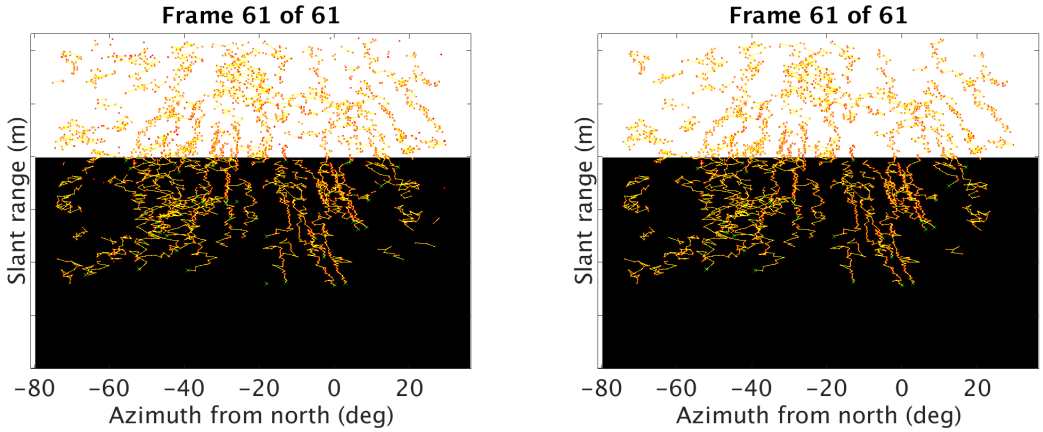


Figure 43: Tracks found in the Östergötland dataset by the SURF keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

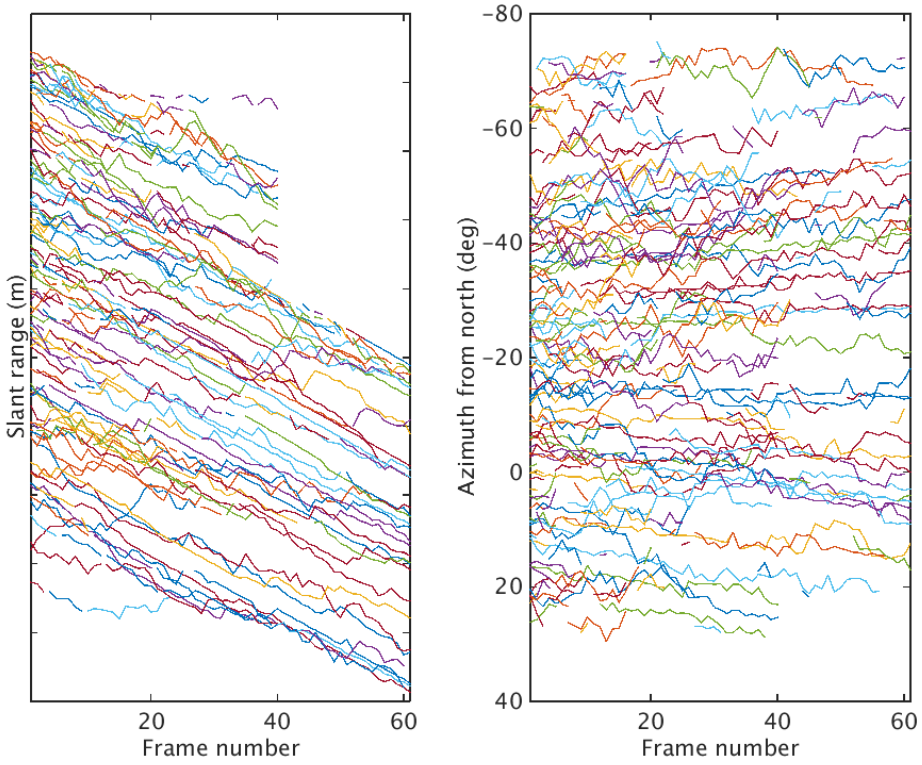
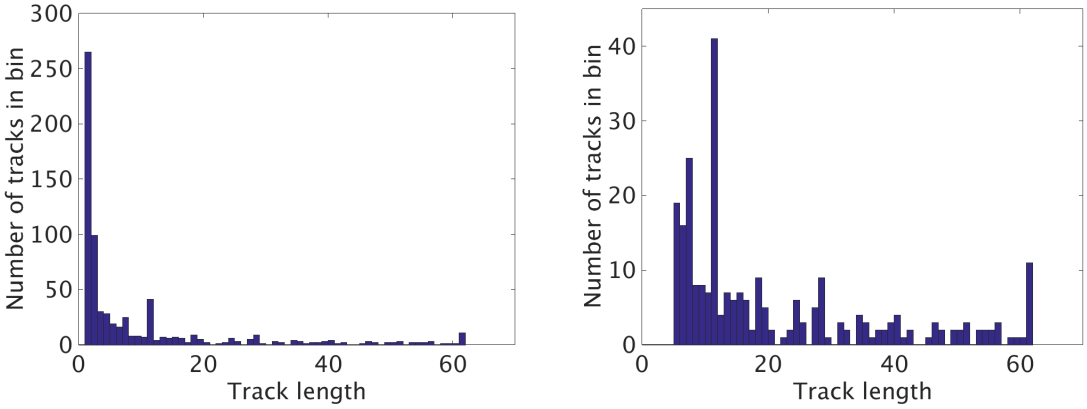


Figure 44: Tracks found in the Östergötland dataset by the SURF keypoint tracking algorithm, separated into slant range and azimuth tracks.



(a) *(b)*

Figure 45: Histograms of the lengths of the found tracks in the Östergötland dataset by the SURF keypoint tracking algorithm. (a) All detected keypoints and tracks. (b) Those tracks that are 5 frames or more in length.

8 Discussion

In this section, a discussion of the results in the previous section will follow. In that section, phenomena clearly apparent in the results from each experiment were mentioned. Here will be a more exhaustive evaluation along with conclusions that can be drawn by combining results from different experiments.

8.1 Scan conversion

Looking at the results from section 7.1.2, we can draw the conclusion that bilinear interpolation is enough. Both interpolation schemes produce visually equivalent results, but bilinear does it in less time. This is because the resolution in the input data is low enough for bicubic interpolation to not contribute any more to the result. Bicubic interpolation would most probably give better results if the input data was better, for example Raven ES-05 data.

The scan conversion algorithm as a whole seems to work well, both on the example image presented in section 7.1.1 and on the radar datasets. Since one frame takes several seconds for the radar to capture, and scan conversion runs in less than one tenth of a second when using bilinear interpolation, the conclusion can be drawn that even this rather naïve implementation in Matlab seems fast enough. However, in the new Raven ES-05 radar, the resolution will be higher, which will make the calculations heavier.

8.2 Denoising

From the results in section 7.2, we can see that the sigma filter is clearly superior to the median filter. However, as in all image processing it is hard to find objective measures of the improvement. Often, one would have to resort to trial and error in order to evaluate the results, which was also done when using it as a preprocessing stage for keypoint tracking. There are hundreds of different filters for noise reduction in images, each with its own advantages and disadvantages, such as noise suppression, edge preservation and execution time. The sigma filter is easy to implement and seems to be "good enough" for the tested data, but there may be better methods for noise suppression in higher resolution data.

8.3 Height estimation

The height estimation algorithm seems to work well when given measurements with low errors. As expected, a larger number of good tracks of points in the terrain leads to better DEM estimation, as can be seen in section 7.3.1.1.

Generally, the worst convergence rates are seen in easting and altitude. The results are better when flying along a curved trajectory than a straight one. It is interesting

to note that convergence along the northing direction is best when flying straight. On the other hand, when flying in a curved trajectory, the northing estimates seems to get slightly worse, while there is a slight improvement in easting convergence. Judging from these results, varying the measurement positions in one direction gives better estimates in that direction. In general, the overall results are better for curved trajectories than straight ones. The altitude estimates also vary a bit when trying different trajectories, but these are dependent on the other two estimates. If the northing or easting error for a peak in the terrain is large, the altitude error is also large since the estimated peak is displaced from the real peak.

When instead varying the altitude, it is apparent that the convergence is worse when flying at lower altitude. This does not feel intuitive at first glance, but consider two points with the same northing and easting position, one at altitude h_1 and one at altitude h_2. When the aircraft is at high altitude, the slant ranges to these points will be distinct from each other. On the other hand, when the aircraft is at low altitude, the slant ranges to these points are closer to each other. In the extreme case, when the aircraft altitude is between h_1 and h_2, the slant ranges to these points can be equal, which means they lie on the same circle. An illustration of this is shown in figure 46.

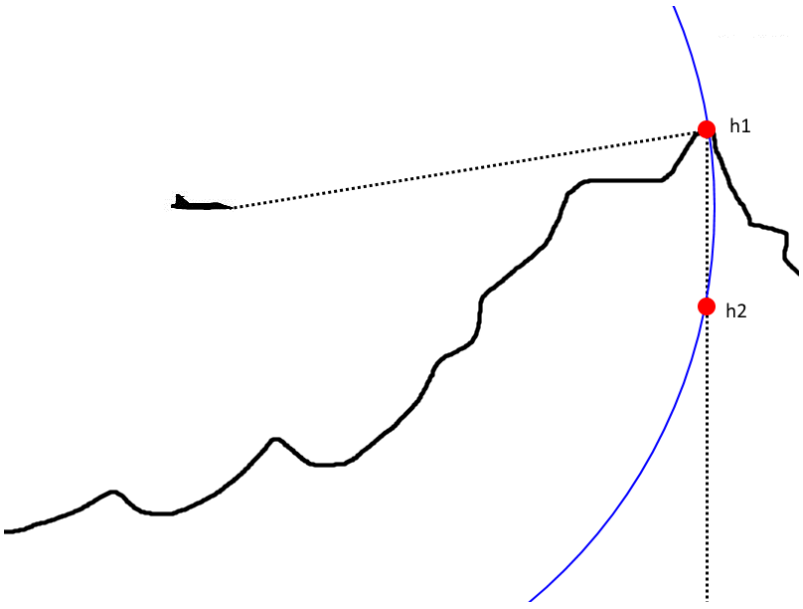


Figure 46: When flying at lower altitudes there is an increasing ambiguity in the height estimation. The extreme case where both points are at equal slant range from the aircraft is illustrated in this figure.

Varying track length gives the expected results that longer tracks give better convergence. Since the height estimation is done by an optimization algorithm, this is natural. The purpose of such an algorithm is to converge when given enough input data.

When combining the above experiments, the convergence is affected in the expected way. Flying at high altitude with long tracks gives good results, while low altitude and short tracks gives horrible results. This superposition also works when mixing

good and bad parameters, for example, flying at high altitude with short tracks gives slightly better results than low altitude with short tracks, and so on. Since there are so many combinations to try, only the fundamental results from varying one parameter at a time were included.

Introducing noise into the measurements, as in section 7.3.2, the convergence properties are significantly worsened. For 50% of the points to converge, the tracks have to be about 60 frames long for the lowest considered noise levels, and even longer for higher noise levels. Tracks of this length are hard to obtain, as seen in the experiments presented in section 7.4.

8.4 Keypoint tracking

Looking at the results from the keypoint tracking experiments in section 7.4, we see that Harris corner detection with digital image correlation works better than the SURF approach. The reason for this is that the points are detected at large scale when using the SURF approach, which gives low accuracy in both range and azimuth. Because of this, the Harris algorithm is considered better for the available data, and therefore those results will be discussed further. However, the tracks are longer when using the SURF algorithm, which means we can not disqualify it altogether. For higher resolution data, such as the Raven ES-05 data, it is still plausible that this algorithm could work better than the Harris algorithm.

A problem is that flying at high altitude gives better convergence, while at the same time lower altitude should make terrain features easier to detect. This is because the radar has better angular resolution when it is closer to the terrain, but there should also be more visible radar shadows which would facilitate keypoint tracking. Layover and foreshortening should both decrease the slant range accuracy in keypoint tracking. Furthermore, the longest tracks are lakes and coastline, which are all typically situated at low altitude, and thus will not contribute much to the DEM estimation.

Since the lengths of the detected tracks are mostly below 20 frames, and judging from the experiments in section 7.3.2, we can draw the conclusion that it is not worth trying to do height estimation using the available datasets. Even at the lowest considered noise levels, a track length of 20 frames would give below 10% convergence. Considering that points are considered converged when they are less than 100 meters away from their correct positions, these tracks are unusable in practice.

However, the datasets are suboptimal, since they are short and contain terrain that is quite flat. Furthermore, they are recorded at high altitude, except the Gotland dataset. However, for that dataset, the terrain is very flat, which makes it less useful. If datasets of hilly terrain recorded from low altitude were available, the results would probably be slightly better, but it is unlikely that this improvement would be large enough for successful height estimation using PS-05 data.

If we consider 50% convergence to be the absolute lower bound for an even remotely

useful DEM and look at the results in section 7.3.2, we can see interesting things. For track lengths of 100 frames, 50% convergence can be reached even for the highest of the considered noise levels. If tracking can be done with higher accuracy, and thus less noisy measurements, much shorter tracks are needed. Such tracking could perhaps be achieved using Raven ES-05 data. Another idea is to reduce noise using temporal filtering instead of spatial. This could probably be done by transforming subsequent frames as if they were captured from the same position using geometrical calculations, and then averaging each pixel in time. Such an approach should be able to conserve small details in the data, while still effectively suppressing noise.

9 Conclusion

Two different approaches to keypoint tracking have been implemented and evaluated. The Harris approach is considered better for the available data, but the SURF approach might prove useful for higher resolution Raven ES-05 data. An already existing algorithm for height estimation has been expanded to allow curved aircraft trajectories, and evaluated on simulated data. Furthermore, auxiliary algorithms for geographical conversion and denoising have been implemented.

In the beginning of this report, the following questions were stated:

- Is it possible to create DEMs using the available RBGM data from the old radar?
- Which demands are there on data resolution and aircraft behaviour in order to obtain DEMs of sufficient precision? Is it plausible to use data from the new radar for this purpose?

Regarding the first question, the answer is that it is impossible using the available datasets, and highly unlikely even if better PS-05 datasets were found. The tracks of sufficient length are too few and mostly correspond to lakes and coastline, which will not contribute much to the estimated DEM.

The answer to the second question is more interesting. Height estimation results are better when flying in a curved trajectory than in a straight one, since this gives more variations in the measurements, leading to better convergence. More varied trajectories are therefore preferable, but they should not be too irregular, since that would probably make keypoint tracking harder. Regarding data resolution it is hard to give an exact answer, but it does not seem impossible to obtain useful results if data resolution is high enough to give more exact tracks, or if tracks can be maintained long enough.

10 Future work

If more time or better datasets were available, there are a number of things that would be interesting to consider:

- Test the algorithms on more suitable datasets, recorded from lower altitude over more hilly terrain.
- Try denoising using temporal filtering as described in the end of section 8.4, in order to suppress noise without removing subtle details in the radar data.
- Use smaller search regions for digital image correlation if higher resolution data were available. Some kind of prediction of where the keypoints are moving would also make it possible to use smaller search areas.
- If the algorithms prove useful on higher resolution data, the height estimation algorithm would have to be expanded to take the curvature of the earth into account, as opposed to assuming that the earth is flat in the considered area.
- Maybe try completely new tracking algorithms for the high resolution data from Raven ES-05.
- Implement the code in C or C++ in order to run it on the actual hardware of the aircraft, to evaluate whether it can run in real time or not.

References

[1] Chris Oliver and Shaun Quegan. *Understanding Synthetic Aperture Radar Images*. Corrected reprint. SciTech Publishing Inc., 2004.

[2] Jung Hum Yu, Xiaojing Li, and Linlin Ge. "Radargrammetric DEM generation using Envisat simulation image and reprocessed image". In: *International Geoscience and Remote Sensing Symposium*. IEEE, 2011, pp. 2980–2983.

[3] Thomas Lillesand, Ralph W. Kiefer, and Jonathan Chipman. *Remote Sensing and Image Interpretation*. 6th edition. John Wiley and Sons Inc., 2007.

[4] Maj. Gregory C. Clark. *Deflating British Radar Myths of World War 2*. Air Command and Staff College, Maxwell Air Force Base. Mar. 1997.

[5] James Clerk Maxwell. "A Dynamical Theory of the Electromagnetic Field". In: *Philosophical Transactions of the Royal Society of London*. Vol. 155. The Royal Society, 1865, pp. 459–512.

[6] Heinrich Hertz. "Ueber Inductionserscheinungen, hervorgerufen durch die electrischen Vorgänge in Isolatoren". In: *Annalen der Physik* 270.6 (1888), pp. 273–285.

[7] Christian Hülsmeyer. "The Telemobiloscope". In: *Electrical Magazine* 2 (1904), p. 388.

[8] Byron Edde. *Radar: Principles, Technology, Applications*. Prentice-Hall Inc., 1993.

[9] Merrill I. Skolnik. *Radar Handbook*. 2nd edition. McGraw-Hill Inc., 1990.

[10] George W. Stimson. *Introduction to Airborne Radar*. 2nd edition. SciTech Publishing Inc., 1998.

[11] *Primed for Supremacy, PS-05/A MK4 Fighter Radar*. Version 1. Saab AB. Apr. 2015.

[12] *Raven ES-05, Active Electronically Scanned Array (AESA) Fire Control Radar*. Selex ES. 2013.

[13] Saab AB. *Gripen NG*. URL: www.saab.com/air/gripen-fighter-system/gripen/gripen/the-fighter/Gripen-NG/ (visited on Mar. 2, 2015).

[14] Canada Centre for Remote Sensing. *Radar Image Distortions*. 2013. URL: www.nrcan.gc.ca/earth-sciences/geomatics/satellite-imagery-air-photos/satellite-imagery-products/educational-resources/9325 (visited on Mar. 5, 2015).

[15] John P. Snyder. *Map Projections - A Working Manual*. Geological Survey professional paper 1395. U.S. Government Printing Office, 1987.

[16] *Gauss Conformal Projection (Transverse Mercator)*. Lantmäteriet, Geodetiska utvecklingsenheten. Aug. 2008.

[17] Jonas Ågren. *Beskrivning av de nationella geoidmodellerna SWEN08_RH2000 och SWEN08_RH70*. Rapportserie: Geodesi och Geografiska informationssystem. Lantmäteriet, Jan. 2009.

[18] European Space Agency. *The Earth's gravity field (geoid) as it will be seen by GOCE*. URL: www.esa.int/spaceinimages/Images/2004/10/The_Earth_s_gravity_field_geoid_as_it_will_be_seen_by_GOCE (visited on June 1, 2015).

[19] *Ytterligare information om ellipsoider och tredimensionella referenssystem*. Lantmäteriet, Geodetiska utvecklingsenheten. May 2012.

[20] Marcin Ligas and Piotr Banasik. "Conversion between Cartesian and geodetic coordinates of a rotational ellipsoid by solving a system of nonlinear equations". In: *Geodesy and Cartography* 60.2 (2011), pp. 145–159.

[21] Toshio Fukushima. "Transformation from Cartesian to geodetic coordinates accelerated by Halley's method". In: *Journal of Geodesy* 79 (2006), pp. 689–693.

[22] Lantmäteriet. *SWEREF 99, projektioner*. URL: www.lantmateriet.se/sv/Kartor-och-geografisk-information/GPS-och-geodetisk-matning/Referenssystem/Tvadimensionella-system/SWEREF-99-projektioner/ (visited on Mar. 18, 2015).

[23] Siddhartha Sikdar et al. "Programmable Ultrasound Scan Conversion on a Mediaprocessor-based System". In: *Proceedings of SPIE*. Vol. 4319. SPIE, 2001, pp. 699–711.

[24] Richard Szelinski. *Computer Vision: Algorithms and Applications*. Electronic draft. Springer Publishing Company, Sept. 2010.

[25] Jong-Sen Lee. "Digital Image Smoothing and the Sigma Filter". In: *Computer Vision, Graphics and Image Processing* 24 (1983), pp. 255–269.

[26] Chris Harris and Mike Stephens. "A Combined Corner and Edge Detector". In: *Proceedings of the Alvey Vision Conference*. Alvey Vision Club, 1988, pp. 23.1–23.6.

[27] Bing Pan, Huimin Xie, and Zhaoyang Wang. "Equivalence of digital image correlation criteria for pattern matching". In: *Applied Optics* 49.28 (Oct. 2010), pp. 5501–5509.

[28] Christopher Evans. *Notes on the OpenSURF Library*. University of Bristol. Jan. 2009.

[29] Lantmäteriet. *Data from GSD-Höjddata, grid 2+*. Obtained through FUK agreement I2014/00579.

[30] *Produktbeskrivning: GSD-Höjddata, grid 2+*. Version 2.0. Lantmäteriet. Feb. 2015.

[31] Lantmäteriet. *Fakta om laserskanning*. URL: www.lantmateriet.se/sv/Kartor-och-geografisk-information/Hojddata/Fakta-om-laserskanning/ (visited on Mar. 10, 2015).

[32] Fusun Balik et al. "Validation of radargrammetric DEM generation from RADARSAT images in high relief areas in Edremit region of Turkey". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. 34. Part XXX.