Data processing of remotely sensed airborne hyperspectral data using the Airborne Processing Library (APL): Geocorrection algorithm descriptions and spatial accuracy assessment

Mark A. Warren, Benjamin H. Taylor, Michael G. Grant, Jamie D. Shutler

Plymouth Marine Laboratory, Prospect Place, Plymouth, PL1 3DH.
Tel: +44 (0)1752 633432 Fax: +44 (0)1752 633101

Abstract

Remote sensing airborne hyperspectral data are routinely used for applications including algorithm development for satellite sensors, environmental monitoring and atmospheric studies. Single flight lines of airborne hyperspectral data are often in the region of tens of gigabytes in size. This means that a single aircraft can collect terabytes of remotely sensed hyperspectral data during a single year. Before these data can be used for scientific analyses, they need to be radiometrically calibrated, synchronised with the aircraft's position and attitude and then geocorrected. To enable efficient processing of these large datasets the UK Airborne Research and Survey Facility has recently developed a software suite, the Airborne Processing Library (APL), for processing airborne hyperspectral data acquired from the Specim AISA Eagle and Hawk instruments. The APL toolbox allows users to radiometrically calibrate, geocorrect, reproject and resample airborne data. Each stage

Email address: `mark1@pml.ac.uk` (Mark A. Warren)

Preprint submitted to Computers and Geoscience *November 11, 2013*

of the toolbox outputs data in the common Band Interleaved Lines (BIL) format, which allows its integration with other standard remote sensing software packages. APL was developed to be user-friendly and suitable for use on a workstation PC as well as for the automated processing of the facility; to this end APL can be used under both Windows and Linux environments on a single desktop machine or through a Grid engine. A graphical user interface also exists. In this paper we describe the Airborne Processing Library software, its algorithms and approach. We present example results from using APL with an AISA Eagle sensor and we assess its spatial accuracy using data from multiple flight lines collected during a campaign in 2008 together with in-situ surveyed ground control points.

Keywords: airborne remote sensing, geocorrection, georectification

1. Introduction

Remote sensing is an established area of science that can be used to capture information over large, potentially hazardous regions. Earth observation remote sensing is usually performed using systems borne on satellites or aircraft, the first such satellite systems going into orbit in the 1970s. The spatial coverage of earth observation instruments tends to be large (in some cases over 1000 square kilometres (km) per scene), and with an increase in spatial and spectral resolutions the volume of data collected can run into terabytes per instrument per year. This is the case for modern, high resolution airborne remote sensing instruments, and it is important to be able to process such data volumes in a timely and efficient manner.

Aircraft remote sensing is of particular importance for many reasons: it allows both testing and calibration of expensive satellite systems before they are launched (Baum et al., 2000) and after launch (Magruder et al., 2010); environmental monitoring (Petchey et al., 2011) with rapid deployment capability with high temporal resolution for hazard mapping (Leifer et al., 2012) and as supporting data for other scientific studies (e.g. Neill et al. (2004)). In Europe and North America alone there are many agencies that use airborne remotely sensed data to derive important information about the Earth's environment. Examples include the US National Oceanic and Atmospheric Administration, NASA, European Space Agency, UK Environment Agency, the UK Natural Environment Research Council (NERC) and the German Aerospace Centre (DLR). Typically these organisations fly with multiple sensors on board, including both passive (such as thermal or hyperspectral scanning instruments) and active (such as lidar or radar). The large spectral and spatial coverage of airborne remotely sensed data can have many uses including: land classification (Liew et al., 2002), vegetation identification (Cochrane, 2000), habitat monitoring (Kooistra et al., 2008), algal bloom detection (Hunter et al., 2010), mineral identification (Crosta, 1996), pollution monitoring (Horig et al., 2001) and geological mapping (Kruse, 1998).

The UK NERC Airborne Research and Survey Facility (ARSF) operates an aircraft that collects remotely sensed data which is disseminated for research use. Two of the instruments are hyperspectral scanners, the Eagle and Hawk, manufactured by Specim Spectral Imaging Ltd. (Specim, 2012). Data collected from each instrument on a single flight mission can result in very large raw data sets of the order of 200 GB, although on average the size is 60-80 GB.

To accomplish efficient data processing, the Airborne Processing Library (APL) has been developed by the ARSF Data Analysis Node based at Plymouth Marine Laboratory (PML). This paper shall discuss the rationale behind APL and how it is exploited within the computing systems at PML including use on a multi-node Grid engine. The processes applied to the hyperspectral data will be introduced and some of the algorithms employed, in particular those for the geocorrection and resampling components, will be discussed in detail. The paper finishes with a look at some example data processing and an analysis on the geocorrection accuracy of a sample data set.

2. Airborne Hyperspectral Data Processing

Typically, remote sensing data requires two broad stages of pre-processing before it is usable for many topics of research. These are: data calibration (Ahern et al., 1987) and data resampling (Toutin, 2004). To compare information collected by different sensors, by different methods, at different locations or at different times the data must be calibrated in some way (Ahern et al., 1987). Typically, remotely sensed data should also be atmospherically corrected to remove scattering due to atmospheric transmission, making them suitable for direct comparison with ground measurements. Atmospheric correction is outside the scope of this paper and is not performed by the APL software. However, the band interleaved by line (BIL) outputs from APL can be imported into existing software such as the ATCOR4 atmospheric correction package (Richter and Schlapfer, 2002). APL outputs in

BIL rather than band interleaved by pixel (BIP) or band sequential (BSQ) as a performance compromise for further processing, since some data users will want to proceed with spatial processing (where BSQ is better suited) and other spectral processing (where BIP is better suited).

Another problem with remotely sensed data is that it may be difficult to analyse without geocorrecting first. For example the captured image is not "North up" or may contain distortions due to platform movements, which can lead to complications when comparing with data from other sources. If this is corrected for, by geocorrecting the data to a well known coordinate system, then it also opens the data up for generation of value-added products. Examples of such being in agriculture and crop management (Seelan et al., 2003) and disaster management (Tralli et al., 2005).

2.1. Pre-development of the Airborne Processing Library

In 2008 an overhaul of the airborne hyperspectral processing chain was proposed so as to improve data processing efficiency and simplify end user interaction. This was initiated with a review of existing software packages for suitability of automated and user-controlled processing. Packages that were considered included the Specim CaliGeo software (Spectral Imaging Ltd, 2004), ENVI software package (Exelis Visual Information Solutions, Boulder, Colorado), ReSe's PARGE (Schlapfer and Richter, 2002) software and the Azimuth System UK's AZ tool package (Azimuth Systems UK, 2005), which in 2008 was the current processing software. No package appeared able to fulfil the requirements of both automated data processing (for example being able to process multiple flight lines without user interaction) and end-user data processing (i.e. simple to understand, licence-free software that can be operated with or without a graphical user interface) - with licensing restrictions for end-users and the inability to freely access the source code being the main disadvantages. The other major disadvantage of the commercial packages is the long term maintenance and security, for example changes in licensing conditions and cost or discontinued support for specific features. Another important factor is transparency, being able to see what is actually being done to the data. Further requirements were being able to react instantly to software bugs and glitches, as well as being able to actively improve and enhance the processing method. With these in mind, having access to source code would be vital for this and played a large factor in the decision to develop APL, which could be tailored for use for both internal, automated processing and end-user data processing.

2.2. Airborne Processing Library

The Airborne Processing Library was developed with a dual remit; to allow quick and efficient processing of the raw hyperspectral data and as a simple, easy to use toolbox for end-users of the data. To reach these goals it was important that the software adhered to the following points:

- Used under Linux operating systems with minimal human interaction
- Used under Windows operating systems
- Include a graphical user interface (GUI)
- Easy to maintain code base

To this end APL has been written using standard C++ (with an optional Python GUI) using minimal third party libraries so as to make cross platform building as simple as possible. Third party libraries involved are the PROJ4 API (PROJ4, 2009) for coordinate re-projections and Blitz++ (Blitz++, 2005) for matrix calculations. All executables are built, from the same source directory, using a desktop PC running Linux (Fedora) using the GNU gcc or mingw-gcc compilers (with the code being portable to other compilers). The GUI has been written to operate on Python version 2.7 using the wXpython graphical libraries. The APL software source code is available to download from: https://github.com/arsf/.

3. Processing Chain

This section describes the data processing chain that employs the APL software. Figure 1 shows a flow diagram of the processing chain including the name of the software utility that performs each action. Details for each action are given in the next sections.

[Figure 1 here.]

3.1. Prior Information

Some information employed in the processing chain exists prior to most data processing and is explained in this section.

- Boresight Correction: this is the angular offset between nadir and the true sensor look direction and is estimated at the start of the flying season and each time the sensors are taken out and replaced into the aircraft, using flight lines which have been collected in a suitable calibration pattern.

- Instrument Calibration: pre- and post-season the hyperspectral sensors go through a rigorous spectral and radiometric calibration to derive a per-pixel gain file and identify spectral wavelength per band. See Choi (2011) and Taylor et al. (2012) for further details including smear correction, stray light and linearity.

- Digital Surface Model (DSM): required to get the best geocorrection accuracy. A DSM is not strictly required as APL will default to an ellipsoid surface, but for hilly and mountainous terrain especially, processing without a DSM will result in large georeferencing errors.

3.2. Radiometric Calibration

The raw data need to be calibrated to give meaning to the values and allow comparisons to other data. This procedure starts by normalising the data using "dark" values - data collected with the shutter closed. This removes noise due to electrical and system components (Oppelt and Mauser, 2007). The data are then scaled using gains calculated during the instrument calibration. A separate mask file is created that contains information on the quality status of each pixel and can be used at a later stage to mask the calibrated data.

3.3. Navigation Synchronisation

The aircraft GPS position and inertial measurement unit (IMU) attitude are post-processed to get a more accurate and smoother solution. This will usually employ a carrier phase differential GPS method (Hoffman-Wellenhof et al., 2001) using the NovaTel GrafNav software together with Leica IPAS software to create a blended IMU/GPS solution. This post-processed navigation data must be synchronised to the image data by comparing instrument and GPS time stamps, using spline interpolation to produce per scan line position and attitude estimates.

3.4. Masking

The optional masking step allows data which have been adversely affected during collection or calibration to be masked out (set to zero) so as not to be used in later scientific analyses. These could be pixels that are over-saturated, pixels that have negative values after dark current subtraction, pixels identified as poorly performing during sensor calibration, pixels identified (by eye) as bad during quality checks, pixels affected due to the smear correction of the Eagle sensor or entire missing scan lines.

3.5. Georeferencing

The georeferencing stage is concerned with computing a per-pixel latitude and longitude map for the image. This is described in detail in section 4.1.

3.6. Re-projection

The optional re-projection phase of the processing transforms the longitude and latitude data into a specified coordinate system (e.g. Universal Transverse Mercator). This is performed using the open source PROJ4 API library, which currently supports more than 120 projections and 42 ellipsoid models.

3.7. Image Georectification

The final stage of the processing is to apply the geocorrection to the radiometrically calibrated data and resample to the desired grid. This is described in detail in section 4.2.

3.8. Automated Processing

The airborne data processing at PML is performed using the Open Grid Scheduler, where individual jobs are dispatched to particular computing nodes on the network for serial batch processing. Each job is formed of the full chain from radiometric calibration through to image resampling. After the initial processing directory is set up no user interaction is required during the processing, until the visual quality inspection of the final results. If jobs need to be resubmitted, for example to correct possible timing errors in the navigation synchronisation, then this is a simple task of editing a text configuration file. In practice each job is submitted with a range of timing offsets to apply to the navigation. This means the radiometric calibration need only be performed once with the subsequent processing stages being performed for each time offset.

To illustrate the processing overheads and storage requirements, a recently collected data set from 2012 consisting of 28 lines (14 of Eagle and 14 of Hawk) was processed on the Grid with a single timing offset for each flight line. The mean length of the flight lines processed was 13784 scan lines, which equates to approximately 35 km at a flying speed of 75 metres per second. The raw data amounts to 82 gigabytes (GB) and took a total of 29 hours of processing time to generate 438 GB of processed, resampled data. However, running in parallel on 22 machines took just 4 hours. Each machine is running the Linux (Fedora 17) operating system and has 8 GB of Random Access Memory (RAM) and a Core i3 processor. It should be noted that the PML Grid is in constant use processing various non-related jobs, some of which will take priority over the submitted airborne jobs. A table showing more detailed data can be found in Appendix A. The table shows that there is a wide variation in processing times that is not necessarily linear with increasing line length. Processing two lines, Hawk_8 and Hawk_9, local to a grid node took 23 minutes and 18 minutes respectively, which shows that processing over the PML network can affect processing times by a factor of at least 4 or 5.

4. Algorithm details

This section describes in detail the algorithms used within APL for the georeferencing and georectification components.

4.1. Georeferencing

The georeferencing stage is concerned with assigning a latitude and longitude value to each pixel of the image data. The basic algorithm is shown in Figure 2 and is described below.

[Figure 2 here.]

4.1.1. Input data

The input data to the algorithm consists of the synchronised navigation information, a DSM (if available) and information about the image data and sensor configuration, i.e. view vectors. The navigation data file is an

ENVI compatible binary BIL file with one record per image line. Each record contains a time stamp and the sensor position (in WGS-84 latitude, longitude and altitude) and attitude (roll, pitch and yaw). The sensor position is constructed from the aircraft GPS position and the sensor lever arms - the distance between the GPS antenna and the sensor origin. Similarly, the attitude values also contain sensor boresight corrections.

The DSM is an elevation model that includes the same area as the scene that is to be geocorrected. It is a binary single band BIL file which contains the height values georeferenced to the WGS-84 latitude and longitude geographic projection.

The sensor view vector file contains an angular vector describing the sensor look angle from the centre of each pixel of the image capture device. These have been calculated using the focal geometry of the sensor. The file is again a binary BIL file.

4.1.2. Algorithm

The algorithm follows the general mathematical direct georeferencing model such as described in Muller et al. (2002).

After initial parameter setup and checks on the input data, the algorithm works on a per scan line basis starting with the earliest collected line. The aircraft position is converted from longitude, latitude, height (LLH) into an Earth Centred Earth Fixed (ECEF) Cartesian XYZ value. Next the sensor view vectors and aircraft attitude are used to create look vectors in ECEF XYZ with the origin at the aircraft position. This is demonstrated in Figure 3.

[Figure 3 here.]

If no DSM is used then these view vectors are projected down on to the ellipsoid surface and the intersection point is stored. This is repeated for each sensor look vector of the scan line. Finally, the intersect points are converted to LLH and written out to a BIL file. The algorithm then moves onto the next scan line.

If a DSM is available then the surface is read into memory at the start of the algorithm, cropped to an over estimate of the predicted cover of the hyperspectral data in order to reduce memory usage. The closest-to-nadir-looking vector is detected and used as the start point for the scan line processing, with the processing continuing for each sensor look vector to the starboard of nadir followed by those port of nadir. The aircraft position in (longitude, latitude) is selected as a 'seed point' for the intersection algorithm as it is assumed that this is close to the nadir view vector intersection. The three nearest DSM points to the seed position are found and a planar surface created, bounded by the 3 DSM vertices. The intersect point between the ECEF XYZ look vector and planar surface is calculated, using basic vector geometry, and if it is contained within the area defined by the 3 DSM vertices then the intersect is stored and the seed point is updated to this new position, ready for the next sensor look vector. If the intersection is outside of the triangle formed by the 3 DSM vertices then 3 new vertices are selected such that they form the opposite triangle which would complete a square. The procedure is repeated and if no intersect is found then the next 3 vertices are selected using a spiral algorithm employed on the seed position such that it is updated as shown in Figure 4. This will be made more efficient in future by deriving the quadrant containing the intersect point (from the look vector direction) and only checking DSM vertices in that quadrant.

The procedure is repeated for each sensor look vector using the updated seed point each time.

[Figure 4 here.]

4.2. Image georectification

The georectification stage is concerned with applying a transformation to the image data and resampling it to a regular grid. The basic algorithm is shown in Figure 5 and is described below.

[Figure 5 here.]

4.2.1. Inputs

The input data required are the outputs from previous stages of the processing. The image data BIL file that is output by the radiometric calibration or masking stage of APL is required. The geolocation file is also required as this contains the pixel location information. To create the output grid it is also required to have information about the desired pixel resolution. Other inputs may be given depending on how the user wishes the georectified image to be created, such as: restricting the output to a particular coverage, selection of image bands to resample, selection of interpolation method to use etc. The output georectified image is an ENVI compatible binary BIL file.

4.2.2. Algorithm

The algorithm has three main steps to it, which can be described as:

- **Restructuring of the input data:** to allow efficient searching of the geolocation file
- **Constructing a Map object:** to define the output image and methods to use for the resampling
- **Creating the resampled image:** perform the resampling and write out the resulting image

The first step is to take the input geolocation data and construct a tree-like structure (called a treegrid from here on), similar to a quadtree, where each node has fixed dimensions rather than number of 'children'. This treegrid groups the points by geographic proximity in order to accelerate neighbourhood searches for the interpolation methods. Figures 6 and 7 show the organisation and conceptual model of the treegrid structure. Since the typical amount of image data is large, in some cases >10 GB, it is not feasible to insert the sensor image data into the treegrid as this is stored in RAM. Instead, only the row and column information describing the pixel location within the data file is inserted into the treegrid. From the row and column indices it is possible to identify both the geolocation and the image data from respective data stores (i.e. files or arrays). Each cell, or node, of the treegrid is known as a 'collection', where each collection has the same fixed size in X and Y, defined by a multiple of the average separation of nadir points. A multiplier of 5 is used as this results in a "middle ground" between the efficient searching within the collections and overheads in searching the treegrid as a whole, with each collection containing approximately 5^2 items. Therefore, for example, if nadir data points are separated by an average of 1 m in the X direction and 2 m in the Y direction, then each collection will have spatial dimensions of 5 m x 10 m.

[Figure 6 here.]

[Figure 7 here.]

The geolocation data file is iterated over and the collection that each pixel belongs to is determined. The information that is inserted into each collection is in the form of an 'item' object. Each item contains the corresponding row and column of the geolocation file, identifying a pixel, and a pointer to an 'ItemData' object, which in turn contains information on where the X, Y geolocation data are stored and methods to read them. When searches are made in the treegrid, all collections within a user-defined radius are searched, to ensure the nearest items are found regardless of which collection contains them.

The second step in the algorithm is to construct a 'Map' object that defines the grid to output data to. This is the main 'work horse' object as it also contains the definitions for interpolating, filling in the grid and writing out the final resampled image. The output grid is constructed based upon the pixel size, the coverage of data (calculated from the tree structure) and the number of bands to output. The Map object also decides how many segments it needs to split the uncorrected image data up into to process efficiently without running low on RAM. By default it allows 1 GB of RAM for holding image data although this can be increased or decreased as the user wishes.

Once this step has completed, the third step of the algorithm is to iterate through each segment in turn, on a row by row basis, and fill the output grid cells with data. By the end of the first segment the full size output file should have been written to disk, zero padded for data yet to be filled in. This allows processing to be done in the order of the uncorrected image data file, irrespective of flight direction or where North is. Further data are inserted on a row by row basis only between the bounds in which the data are contained. For each column of the row to be written, items are found from the tree and passed to the interpolator. The interpolator takes these data and returns the interpolated image value for insertion into the grid. If, however, one of these items contains the 'masked' data value for a band being resampled then it is ignored (for that band only) and the next nearest non-masked item is used. If there are none within the search radius then a value of zero is returned from the interpolator for that band. Further information on the interpolation methods can be found in Appendix B.

5. Results

5.1. Data products

An example of APL processed Eagle data products, for an area over the River Thames in London, can be seen in Figure 8. The Eagle data shown are (a) prior to applying radiometric calibration, (b) after applying radiometric calibration and (c) shows the data after georectification. Also shown in the figure are two spectral plots from the same green vegetation feature, one from the raw data and one from the calibrated and georectified data. As no atmospheric correction has been performed on the data, any effects due to the atmosphere will still remain in the data, where these errors will have a direct effect on the amplitude of the reflectance signal but the general shape of the spectra should be unaffected. In Figure 8(e) it can be seen that the calibrated spectra clearly shows the "red edge" at around 700 nano-metres (nm) that one expects to find in vegetation data. In contrast there are two peaks in the raw uncorrected data (Figure 8(d)) illustrating that uncorrected data cannot be relied upon for spectral information.

[Figure 8 here.]

A second example showing the geocorrection results of APL can be seen in Figure 9. The data in the sensor geometry can be seen at the top in Figure 9(a), and in the main image after georectification into the Ordnance Survey National Grid projection in Figure 9(b). The image background includes Ordnance Survey VectorMap District OpenData to illustrate the geocorrected data. The top left of Figure 9(b) shows a zoomed view to highlight the geocorrection at one of the motorway junctions.

[Figure 9 here.]

5.2. APLCORR Georeferencing analysis

The accuracy of the georeferencing of the data has been tested using hyperspectral data collected in 2008 over a calibration site in Cambridgeshire, UK. The site contains seven GPS surveyed targets which are visible in the image data. Eight flight lines from the Eagle sensor were processed with

APL and the seven targets were identified from the images prior to resampling. The georeferencing output were re-projected into a Universal Transverse Mercator projection (Zone 30) for ease of dealing with errors in metres (m) rather than degrees. Not all GPS control points were visible in each dataset. Figure 10 shows the calibration site with the targets identified. The post-processed navigation solution file contains data at 200 Hz, and the image data is recorded at 40 frames per second. A digital surface model has been used generated from the NEXTMap 5 m resolution product (Intermap Technologies, 2007).

[Figure 10 here.]

Appendix C shows the full dataset. The Easting and Northing errors have been converted to along and across track errors by rotation using the mean heading of the aircraft for each section covering the GCPs for each flight line. The mean absolute along track error from the 7 targets and 8 flight lines (42 samples in total) for the Eagle sensor is 0.74 m $\pm$ 0.58 m. The mean absolute across track error is 0.39 m $\pm$ 0.25 m. We expect larger measurement errors in the along track since the spatial resolution is lower in this direction. At nadir the along track pixel separation is approximately 1.9 m whereas the across track pixel separation is approximately 0.60 m. This would lead us to expect a higher reported error in the along track direction as the centre of the pixel is being used as the identified location, and this is observed in the results. We can take the ratio of the error versus the pixel separation to approximate the error in terms of pixel size, giving the following mean absolute along track error (at nadir): 0.39 $\pm$ 0.31 and across track error (at nadir): 0.65 ± 0.42 reported in pixel size. However, it should be noted that the pixel size will vary along and across track due to the surface topography, aircraft altitude and velocity and target swath position.

6. Conclusions

The Airborne Processing Library (APL) toolbox has been developed and in operational use since 2011. It allows users to radiometrically calibrate, geocorrect, re-project and re-sample remotely sensed optical airborne data. It can be operated on Windows or Linux systems via command line, a graphical user interface (GUI) or through a Grid Engine. The core geocorrection and resampling algorithms have been discussed. The absolute along and across track spatial geocorrection accuracy have been assessed and reported. The reduced along track accuracy is likely due to the lower spatial resolution (larger spatial coverage) of the sensor configuration in this direction. A high spatial accuracy is important when analysing large volumes of data as it allows much easier dataset integration within Geographic Information System (GIS) applications and other tools used for post-processing and analysing such data.

Acknowledgements

The authors would like to thank Dr Peter Land for useful discussions on reflectance spectra of ground targets. Figure 9 contains Ordnance Survey OpenData © Crown copyright and database right 2013. The hyperspectral data used in this report were collected by the Natural Environment Research Council Airborne Research and Survey Facility.

References

Ahern, F. J., Brown, R., Cihlar, J., Gauthier, R., Murphy, J., Neville, R. A., Teillet, P. M., 1987. Radiometric correction of visible and infrared remote sensing data at the Canada Centre for Remote Sensing. International Journal of Remote Sensing 8 (9), 1349–1376.

Azimuth Systems UK, 2005. AZGCORR: User's manual. 57pp URL: http://arsf.nerc.ac.uk/documents/azgcorr_v5.pdf, [Accessed March 2013].

Baum, B. A., Kratz, D. P., Yang, P., Ou, S., Hu, Y., Soulen, P. F., Tsay, S.-C., 2000. Remote sensing of cloud properties using modis airborne simulator imagery during success 1. data and models. Journal of Geophysical Research 105 (D9), 11767–11780.

Blitz++, 2005. Blitz++ library v0.9. URL: http://sourceforge.net/projects/blitz/, [Accessed January 2013].

Catmull, E., Rom, R., 1974. A class of local interpolating splines. In: Barnhill, R. E., Reisenfeld, R. F. (Eds.), Computer Aided Geometric Design. Academic Press, New York, pp. 317–326.

Choi, R. K. Y., 2011. Characterisation of NERC ARSF AISA Eagle and Hawk. In: European Association of Remote Sensing Laboratories (EARSeL) 7th SIG-Imaging Spectroscopy Workshop, Edinburgh, 11-13 April. 57pp URL: http://www.earsel2011.com/content/download/Proceedings/S7_6_YoungChoi_pres.pdf, [Accessed October 2013].

Cochrane, M. A., 2000. Using vegetation reflectance variability for species level classification of hyperspectral data. International Journal of Remote Sensing 21 (10), 2075–2087.

Crosta, A. P., 1996. High Spectral Resolution Remote Sensing for Mineral Mapping in the Bodie and Paramount Mining Districts, California. In: International Archives of Photogrammetry and Remote Sensing (IAPRS), Vol.XXXI, ISSN 1682-1750. pp. 161–166.

Hoffman-Wellenhof, B., Lichtenegger, H., Collins, J., 2001. GPS: Theory and Practice. Springer-Verlag, 382pp.

Horig, B., Kuhn, F., Oschutz, F., Lehmann, F., 2001. Hymap hyperspectral remote sensing to detect hydrocarbons. International Journal of Remote Sensing 22 (8), 1413–1422.

Hunter, P. D., Tyler, A. N., Carvalho, L., Codd, G. A., Maberly, S. C., 2010. Hyperspectral remote sensing of cyanobacterial pigments as indicators for cell populations and toxins in eutrophic lakes. Remote Sensing of Environment 114 (11), 2705–2718.

Intermap Technologies, 2007. NEXTMap Britain: Digital terrain mapping of the UK. NERC Earth Observation Data Centre. URL: `http://badc.nerc.ac.uk/view/neodc.nerc.ac.uk__ATOM__dataent_11658383444211836`, [Accessed October 2013].

Kooistra, L., Mucher, S., Niewiadomska, A., 2008. Monitoring of natura 2000 sites using hyperspectral remote sensing: Quality assessment of field and airborne data for Ginkelse and Ederheide and Wekeromse Zand. Tech. rep., Wageningen University.

Kruse, F. A., 1998. Advances in hyperspectral remote sensing for geologic mapping and exploration. In: Proceedings 9th Australasian Remote Sensing Conference, Sydney, Australia.

Leifer, I., Lehr, W. J., Simecek-Beatty, D., Bradley, E., Clark, R., Dennison, P., Hu, Y., Matheson, S., Jones, C. E., Holt, B., Reif, M., Roberts, D. A., Svejkovsky, J., Swayze, G., Wozencraft, J., 2012. State of the art satellite and airborne marine oil spill remote sensing: Application to the BP Deepwater Horizon oil spill. Remote Sensing of Environment 124, 185–209.

Liew, S. C., Chang, C. W., Lim, K. H., 2002. Hyperspectral land cover classification of EO-1 Hyperion data by principal component analysis and pixel unmixing. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2002. Vol. 6. pp. 3111–3113.

Magruder, L., Ricklefs, R., Silverberg, E., Horstman, M., Suleman, M., Schutz, B., 2010. Icesat geolocation validation using airborne photography. IEEE Transactions on Geoscience and Remote Sensing 48 (6), 2758–2766.

Muller, R., Lehner, M., Muller, R., Reinartz, P., Schroeder, M., Vollmer, B., 2002. A program for direct georeferencing of airborne and spaceborne line scanner images. In: International Society for Photogrammetry and Remote Sensing (ISPRS) Proceedings XXXIV Part 1.

Neill, S., Copeland, G., Ferrier, G., Folkard, A., 2004. Observations and numerical modelling of a non-buoyant front in the Tay Estuary, Scotland. Estuarine, Coastal and Shelf Science 59 (1), 173–184.

Oppelt, N., Mauser, W., 2007. Airborne Visible / Infrared Imaging Spectrometer AVIS: Design, Characterization and Calibration. Sensors 7 (9), 1934–1953.

Petchey, S., Brown, K., Hambidge, C., Porter, K., Rees, S., 2011. Operational use of remote sensing for environmental monitoring. Tech. rep., Natural England/Environment Agency Collaboration, 88pp URL: `cdn.environment-agency.gov.uk/geho1211bvqs-e-e.pdf`, [Accessed January 2013].

PROJ4, 2009. Proj.4 - cartographic projections library - version 4.7.1. URL: `http://trac.osgeo.org/proj/`, [Accessed July 2012].

Richter, R., Schlapfer, D., 2002. Geo-atmospheric processing of airborne imaging spectrometry data. part 2: Atmospheric/topographic correction. International Journal of Remote Sensing 23 (13), 2631–2649.

Schlapfer, D., Richter, R., 2002. Geo-atmospheric processing of airborne imaging spectrometry data part 1: Parametric orthorectification. International Journal of Remote Sensing 23 (13), 2609–2630.

Seelan, S. K., Laguette, S., Casady, G. M., Seielstad, G. A., 2003. Remote sensing applications for precision agriculture: A learning community approach. Remote Sensing of Environment 88 (1-2), 157–169.

Specim, 2012. Specim - spectral imaging ltd. URL: `http://www.specim.fi/`, [Accessed July 2012].

Spectral Imaging Ltd, 2004. SPECIM CaliGeo 4.0..46 AISA+/AISA Eagle Data Processing Tool Operating manual. 37pp.

Taylor, B., Choi, K.-Y., Warren, M., Grant, M., Goy, P., Johnson, J., Panousis, I., 2012. NERC ARSF hyperspectral instruments: calibration procedures, characteristics and effects. In: Proceedings of the Remote Sensing and Photogrammetry Society (RSPSoc) Conference 2012 - Changing how we view the world.

Toutin, T., 2004. Geometric processing of remote sensing images: models, algorithms and methods. International Journal of Remote Sensing 25 (10), 1893–1924.

Tralli, D. M., Blom, R. G., Zlotnicki, V., Donnellan, A., Evans, D. L., 2005. Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards. International Society for Photogrammetry and Remote Sensing (ISPRS) Journal of Photogrammetry and Remote Sensing 59 (4), 185–198.

Appendix A. Processing performance

[Table 1 here.]

Appendix B. Interpolation of treegrid data

There are currently 4 interpolation methods used in the APL resampling:

- Nearest neighbour
- Inverse distance weighted
- Bi-linear

- Cubic

The interpolator takes input from a treegrid search - of which there are two types: 'nearest points' or 'nearest quadrant points'. The difference between the two being that 'nearest points' search just returns the nearest N items to the given search point, ordered by distance, whereas 'nearest quadrant points' returns the nearest N points ordered by quadrant centred on the search point. For example, in Eastings and Northings, using a 'nearest quadrant points' search for one point, will return four points: one to the North-East, one to the South-East, one to the South-West and one to the North-West of the given search point. This search is used for the bilinear and cubic interpolators. The nearest neighbour and inverse distance weighted interpolators use the 'nearest points' search. Graphical representations of the interpolation methods are shown in Figure 11.

[Figure 11 here.]

Appendix B.1. Nearest Neighbour

The nearest neighbour interpolator simply takes the image data value from the nearest item to the search point.

Appendix B.2. Inverse Distance Weighted

The inverse distance weighted method follows the basic Shepard method (Shepard, 1968), defined as:

$$w_i = {distance_i}^{-2} / \sum distance_j^{-2}$$
$$f(x) = \sum w_i * f(i)$$

where w_i are weights and $f(x)$ is the image data value of item x.

Appendix B.3. Bilinear

Bilinear interpolation takes the 4 nearest items (A, B, C and D) to the search point, X, such that the items form a quadrilateral containing the search point (see Figure 12). Using the geolocation information of each item the following formulae can be solved for the scalars U and V:

$$P = A + U * (B - A)$$

$$Q = D + U * (C - D)$$

$$X = P + V * (Q - P)$$

[Figure 12 here.]

The values of U and V, which are within the range 0-1, are then used to weight the item data values in the interpolation formula:

$f(X) = f(A) * (1 - V) * (1 - U) + f(B) * (1 - V) * U + f(D) * (1 - U) * V + f(C) * U * V$

where $f(x)$ is the image data value of cell x.

Appendix B.4. Cubic

Cubic interpolation uses 16 nearest items such that there are 4 in each quadrant surrounding the centre of the cell. Using a series of 1-dimensional cubic Catmull-Rom splines (Catmull and Rom, 1974) these data are interpolated. The final result is obtained by interpolating with 4 splines in the X direction followed by 1 spline in the Y direction.

Appendix C. Geocorrection analysis results

[Table 2 here.]

List of Figures

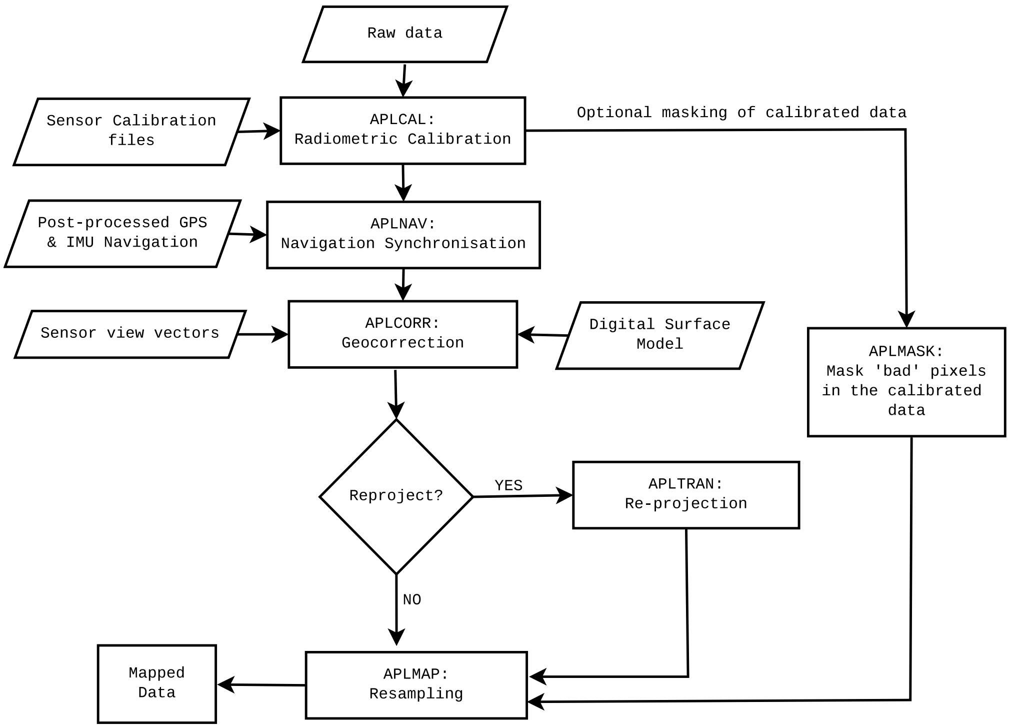

Figure 1: Flow diagram of the hyperspectral processing chain.

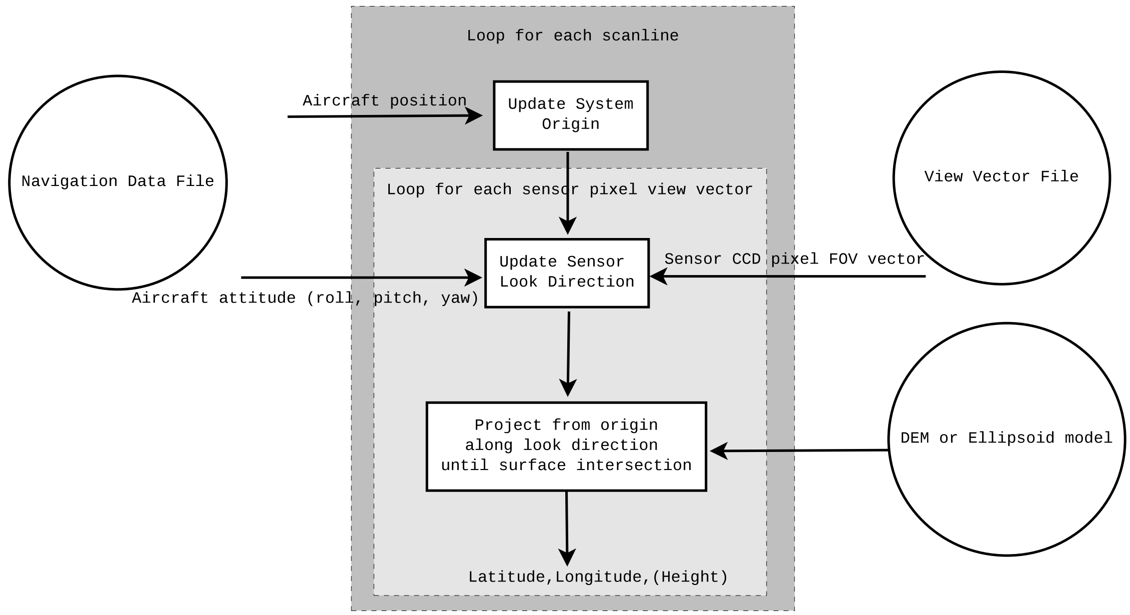

Figure 2: Flow diagram of the APL georeferencing algorithm, where FOV is the sensor field of view.

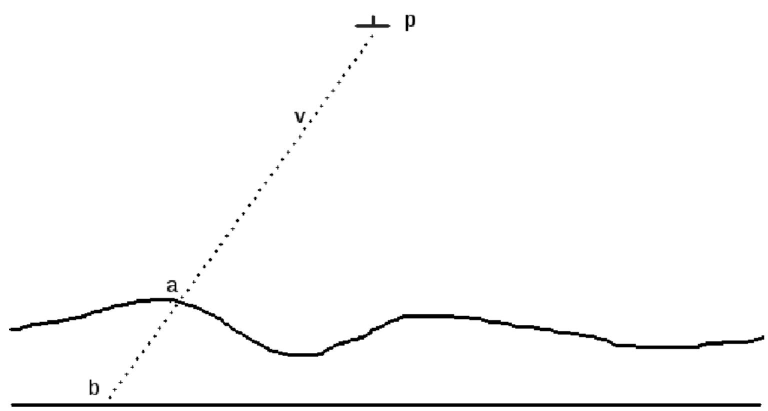

Figure 3: Intersection of view vector to find geolocation of image pixel. Using the position of aircraft p and the sensor view vector v, the intersection point with the surface model can be found. In this example, intersection point a is found when using a DSM whereas intersection point b is found if using the ellipsoid surface model.

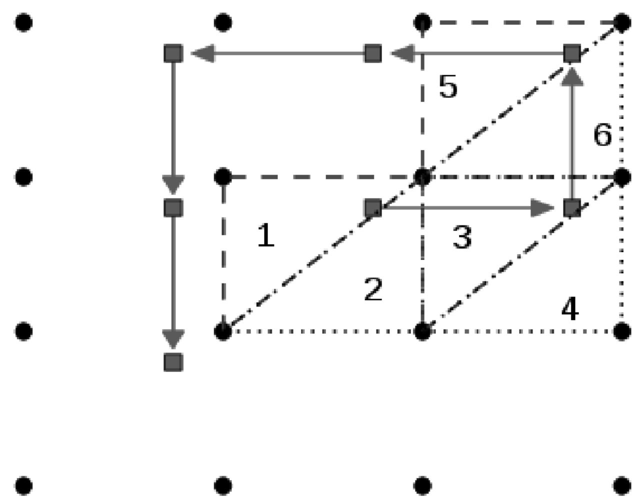

Figure 4: Spiral updating of seed position (square) in the direction of the arrows. Circles represent the DSM vertices. The dashed-line triangles represent the first planar surface to be tested for each seed position, the dotted-line triangles the ‘opposite’ plane that would complete a square. Only the first three sets are shown for clarity, with the triangles numbered in the order of being tested.

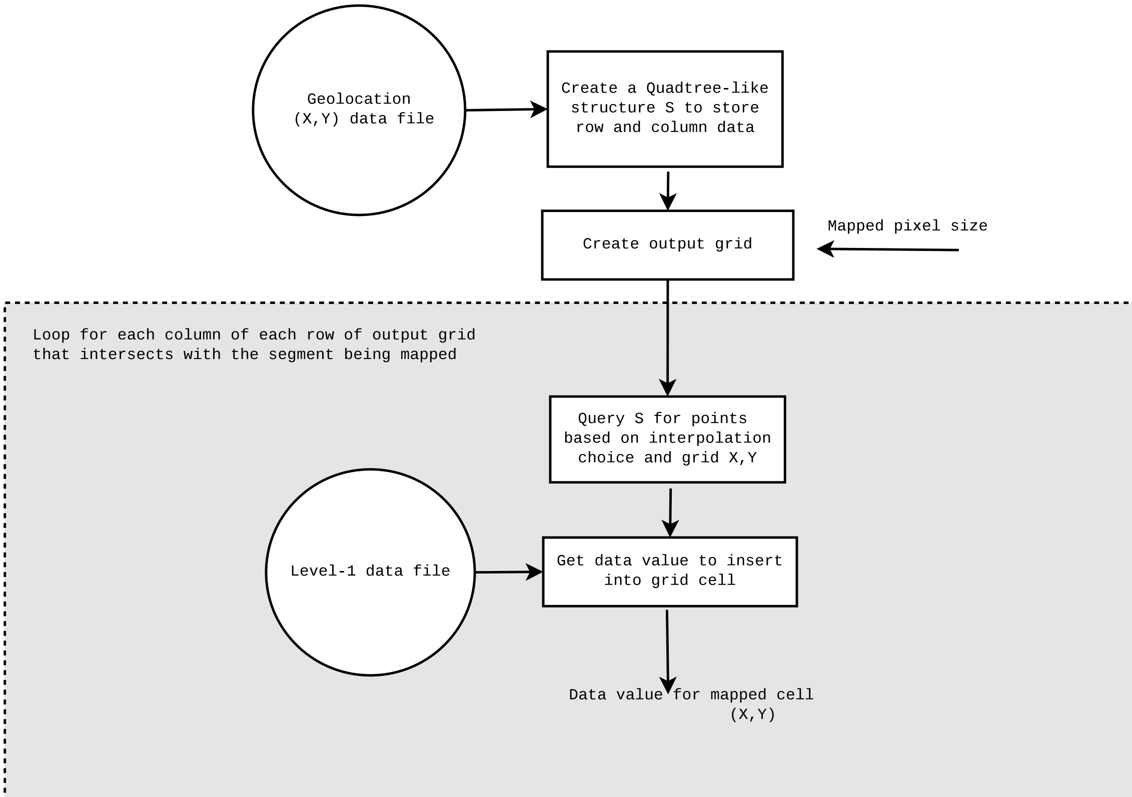

Figure 5: Flow diagram of the APL georectification algorithm.

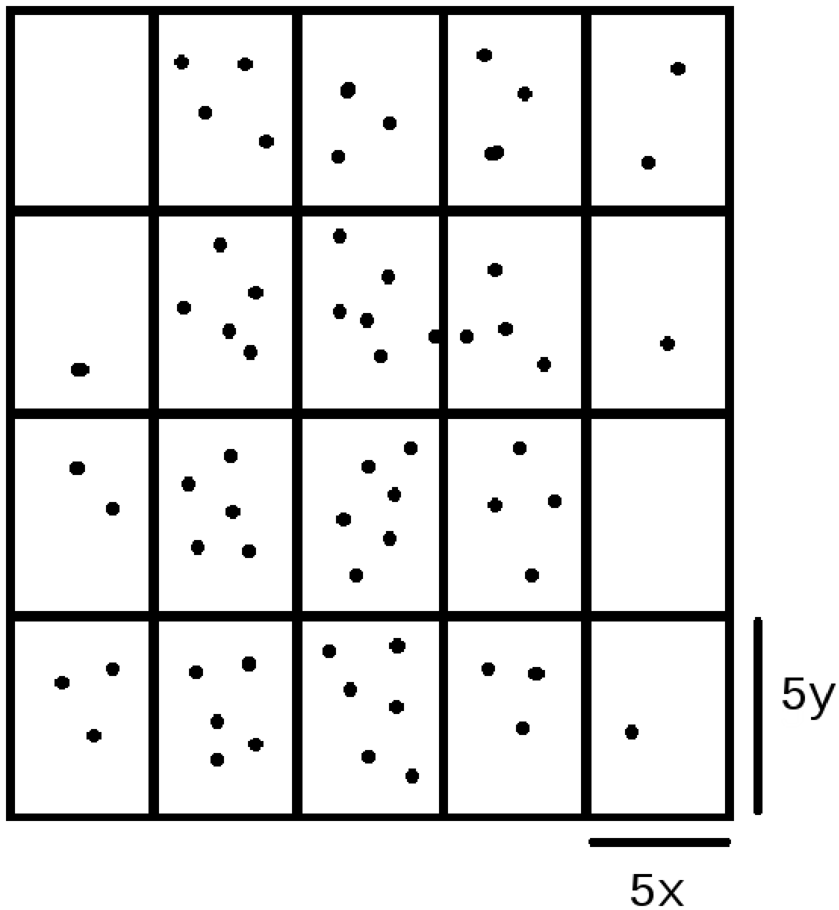

Figure 6: Tree-like structure shown as a 2-dimensional grid overlaying the data points. Each cell of the grid is a ‘collection’ containing the data points, known as ‘items’. Each collection has dimensions in X and Y (e.g. Eastings and Northings) equal to five times the mean spacing of data points at nadir. Items are inserted into the collection which bounds the item X,Y position. This will typically result in 25-30 items per collection at nadir, with fewer items per collection at the edge (the number of items in the diagram have been reduced for simplification).

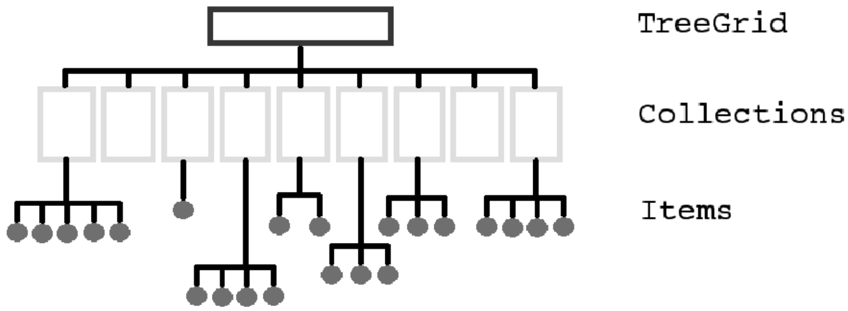

Figure 7: Organisational overview of the treegrid. The treegrid contains a series of collections (defined by geographic region) which in turn contain items (references to image data points). The organisation of data points in a tree like this allows for efficient searching based on the X,Y position.

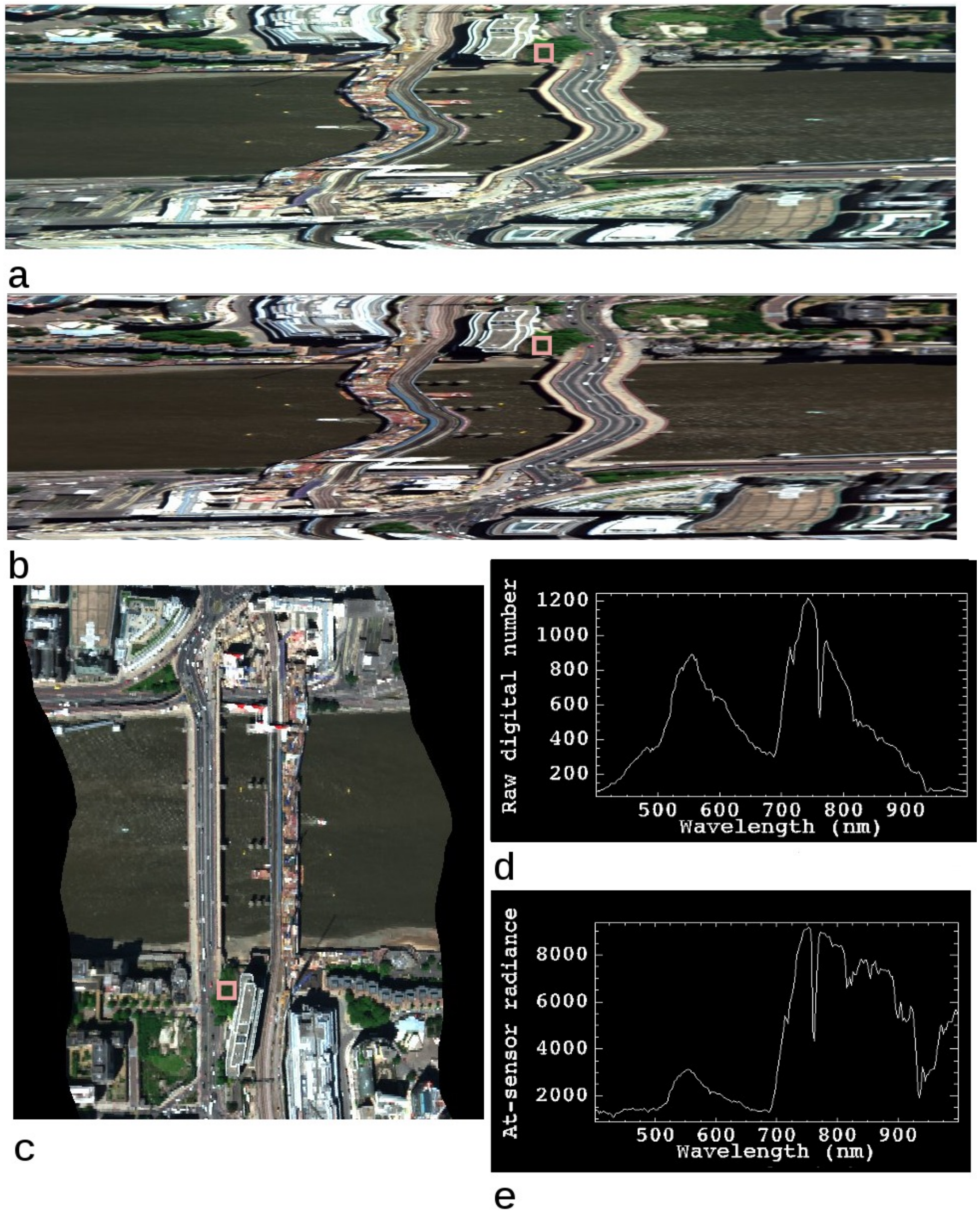

Figure 8: Example Eagle sensor (a) raw data, (b) radiometrically calibrated data and (c) georeferenced and resampled data. Spectral plots of green vegetation in raw and calibrated data have been plotted to show differences in these data, and shown in (d) and (e) respectively. This feature is highlighted in (a), (b) and (c) by a pink square. Note 'red edge' at 700 nm becomes much more apparent in calibrated data than raw data.

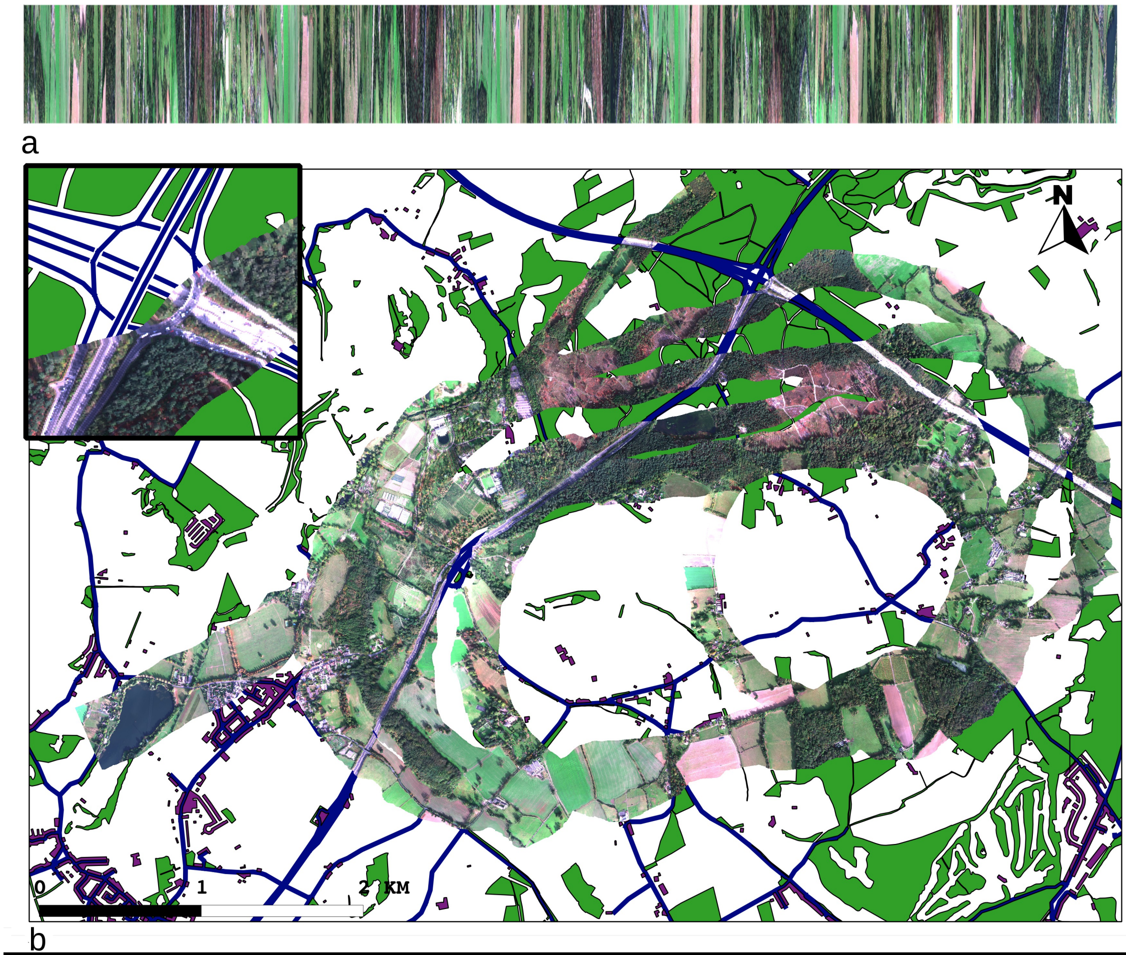

Figure 9: Example Eagle data that are (a) prior to geocorrection and (b) after geocorrection and resampling. Also shown are Ordnance Survey OpenData vectors with roads in blue, woodland in green and buildings in purple. Top left of (b) shows a zoom window of the junction to highlight the geocorrection. Eagle data is a spiral flight line collected near the south west of the M25 motorway in 2011.

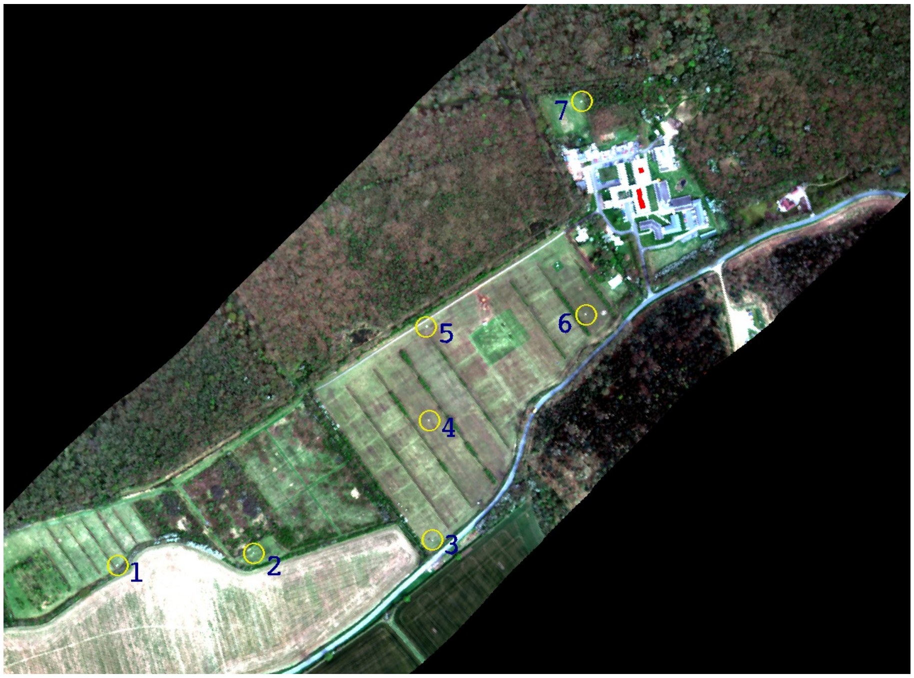

Figure 10: The Monks Wood calibration site Cambridgeshire, UK. The seven surveyed GPS targets are circled and numbered.

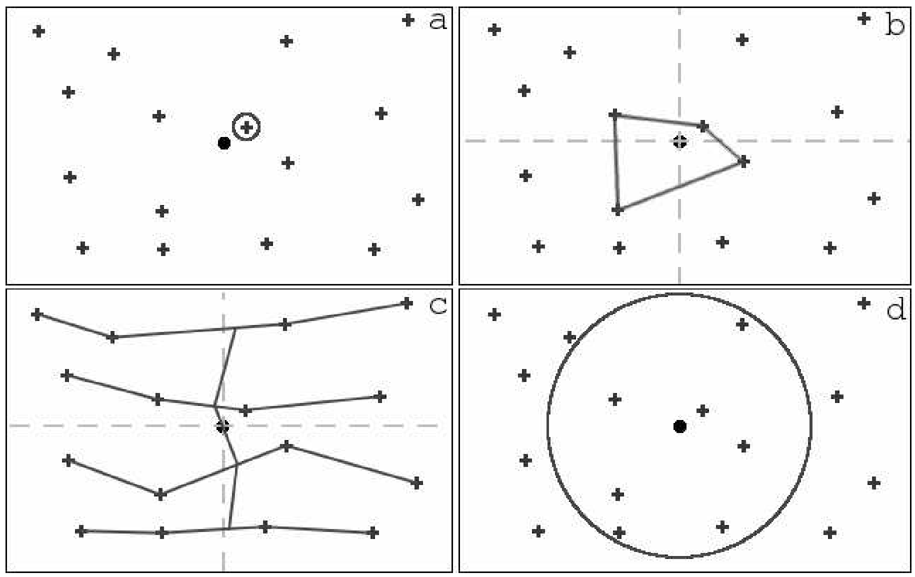

Figure 11: Illustration of the 4 interpolation methods; the filled circle is the cell point to be interpolated and crosses are treegrid items. a) Nearest neighbour interpolation selects the item nearest to the cell to be interpolated. b) For bi-linear interpolation, the nearest item from each quadrant centred on the cell to be interpolated is selected, forming a quadrilateral surrounding the cell. A product of two linear interpolations is performed to determine the interpolated value at the cell. c) Cubic interpolation finds the nearest 4 items in each quadrant centred on the cell to be interpolated. These 16 items are then used to form a series of Catmull-Rom splines to interpolate the value at the cell. d) Inverse distance weighted interpolation finds up to the nearest N items within a search radius and takes a weighted average, where the weights are based on the inverse of the distance of each item from the cell to be interpolated.

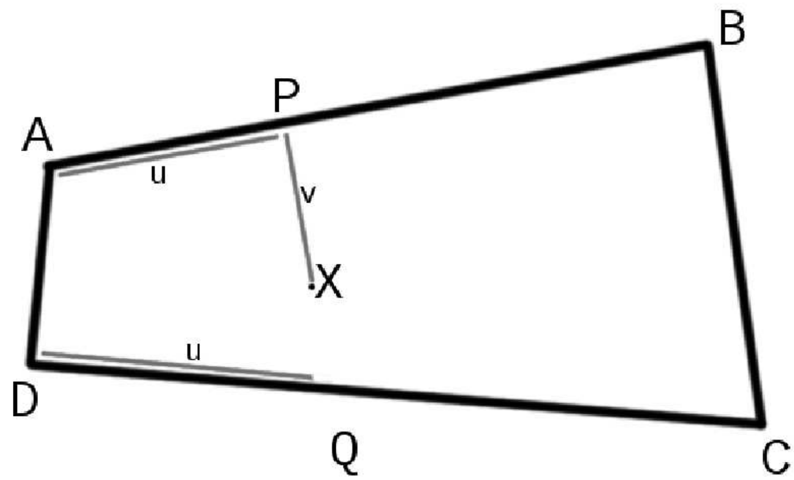

Figure 12: The calculation of the position of point X in terms of U and V based on 4 surrounding points. U and V are scalars which are used to weight the data values in the bilinear interpolation algorithm.

List of Tables

Line	Process time (hh:mm:ss)	Flight length (scan lines)	Number of bands
Eagle_-1	00:26:32	16245	126
Eagle_-2	00:01:18	1881	126
Eagle_-3	01:47:31	15321	126
Eagle_-4	00:32:42	18098	126
Eagle_-5	02:58:43	15646	126
Eagle_-6	01:18:35	16868	126
Eagle_-7	01:15:55	16153	126
Eagle_-8	01:09:05	15693	126
Eagle_-9	00:46:33	13492	126
Eagle_-10	00:56:02	14219	126
Eagle_-11	00:50:24	12323	126
Eagle_-12	00:29:54	12047	126
Eagle_-13	00:34:06	8643	126
Eagle_-14	00:25:03	6909	126
Hawk_-1	01:32:31	16247	233
Hawk_-2	01:31:23	16539	233
Hawk_-3	01:25:32	15322	233
Hawk_-4	01:23:33	18099	233
Hawk_-5	01:22:43	15646	233
Hawk_-6	01:24:46	16868	233
Hawk_-7	01:24:14	16155	233
Hawk_-8	02:00:16	15694	233
Hawk_-9	01:08:51	13492	233
Hawk_-10	00:50:58	14221	233
Hawk_-11	00:21:39	12324	233
Hawk_-12	00:27:45	12049	233
Hawk_-13	00:08:17	8645	233
Hawk_-14	00:05:23	6910	233

Table 1: Table showing processing performance statistics for processing on the Grid.

Flight line	Target	Abs E	Abs N	Abs Along	Abs Across
1	3	0.098	0.334	0.302	0.174
1	4	0.265	0.710	0.682	0.331
1	5	0.467	0.790	0.883	0.249
1	6	0.105	0.436	0.225	0.388
2	3	0.392	0.404	0.439	0.353
2	4	0.465	0.730	0.684	0.531
2	5	0.727	0.400	0.439	0.687
2	6	0.205	1.264	1.278	0.087
2	7	0.355	0.404	0.369	0.391
3	1	1.310	1.765	2.166	0.373
3	2	0.109	0.437	0.223	0.391
3	3	0.558	0.464	0.083	0.721
3	4	0.615	0.170	0.562	0.302
3	5	1.633	1.220	2.024	0.245
3	6	1.375	0.726	1.496	0.424
3	7	1.025	1.456	1.747	0.346
4	1	0.750	0.885	0.764	0.873
4	2	1.621	0.653	1.631	0.627
4	3	0.422	0.576	0.431	0.569
4	4	0.875	0.430	0.882	0.416
4	5	0.197	0.570	0.206	0.567
4	6	0.005	0.546	0.004	0.546
5	3	0.568	1.286	0.534	1.301
5	4	0.395	0.030	0.396	0.020
5	5	2.093	0.270	2.099	0.215
5	6	1.335	0.016	1.334	0.051
6	1	0.250	0.935	0.795	0.552
6	2	0.441	0.017	0.347	0.272
6	3	0.278	0.424	0.062	0.503
6	4	0.045	0.060	0.004	0.075
6	5	0.997	0.150	0.858	0.530
6	6	0.205	0.114	0.230	0.046
6	7	1.105	0.076	0.794	0.772
7	3	0.352	0.936	0.950	0.313
7	4	0.055	0.370	0.323	0.189
7	5	0.183	0.410	0.434	0.115
7	6	0.395	0.264	0.453	0.142
8	3	0.038	1.076	1.071	0.106
8	4	0.385	0.430	0.454	0.357
8	5	0.343	0.180	0.201	0.331
8	6	0.285	0.974	0.990	0.223
8	7	0.445	1.206	1.175	0.521
	Mean	0.566	0.586	0.739	0.386
	St Dev	0.499	0.429	0.579	0.254

Table 2: Absolute errors (in metres) between GPS and target identification from geocorrection data (prior to resampling). Errors reported in Eastings, Northings and converted to along track, across track.