Airborne Vision-Based Mapping and Classification of Large Farmland Environments

Mitch Bryson, Alistair Reid, Fabio Ramos and Salah Sukkarieh
Australian Centre for Field Robotics
University of Sydney, NSW
Australia
{m.bryson,a.reid,f.ramos,salah}@acfr.usyd.edu.au

Abstract

This paper presents a framework for integrating sensor information from an Inertial Measuring Unit (IMU), Global Positioning System (GPS) receiver and monocular vision camera mounted to a low-flying Unmanned Aerial Vehicle (UAV) for producing large-scale terrain reconstructions and classifying different species of vegetation within the environment. The reconstruction phase integrates all of the sensor information using a statistically optimal non-linear least squares bundle adjustment algorithm to estimate vehicle poses simultaneously to a highly-detailed point feature map of the terrain. The classification phase uses feature descriptors based on the colour and texture properties of vegetation observed in the vision data, and uses the terrain information to build a geo-referenced map of different types of vegetation. The resulting system can be used for a range of environmental monitoring missions such as invasive plant detection and biomass mapping. Experimental results of the algorithms are demonstrated in a 'weed-finding' mission over a large farmland area of the Australian outback.

1 Introduction

Environmental monitoring of large areas of wilderness and farmland relies on periodic geo-referenced mapping of the environment in order to detect invasive plants and weeds, tally the quantity of biomass and develop effective environmental management strategies. Traditionally, data in environmental applications has been collected via high-flying surveys using manned-aircraft or through satellite remote sensing. Although these methods have the potential to cover large areas, the resolution of the resulting surveys is often limited (based on the operating altitude of these vehicles) and the cost involved in operations is often very high. There is much interest in the use of low-flying Unmanned Aerial Vehicles (UAVs) as a cost-effective method for creating high-resolution terrain maps over large areas. The advantage in the use of UAVs is that they typically have a longer endurance than human pilots and being smaller can potentially fly at lower altitudes, building higher resolution 3D maps

by 'hugging' the terrain as they fly. Additionally, with the use of autonomous computer vision algorithms, the resulting resolution of maps in these applications has the potential to make them suitable for detecting and identifying a broad class of different types of flora and fauna that would otherwise be difficult to distinguish from higher altitudes.

A distinct challenge facing cost-effective methods for airborne surveying is in the types of sensor data required for producing geo-referenced maps of the terrain and identifying different flora and fauna in the environment. Sensors such as multi-spectral and hyper-spectral imagers are typically used in satellite remote sensing to classify different types of vegetation based on their spectral characteristics. The cost associated with these sensors is high with respect to options such as a commonly available vision camera which is constrained to imaging within the visible spectrum. Although vision sensors are cheap, the constraint of the resolvable spectra reduces the discriminating power this sensor has for distinguishing between different types of vegetation. This can however be overcome in some sense by the increase in resolution of imagery available based on the low-altitude operations of a UAV system; the structure of vegetation can now be resolved which provides some scope in discriminating between different types.

In the case of mapping objects in these types of 3D environments, sensors such as laser range finders have been typically used, particularly by Unmanned Ground Vehicles (UGVs); however a cost-effective system for an airborne platform prohibits the use of these sensors due to the weight and power requirements. In both the tasks of mapping and classification, vision sensors therefore provide the only real low-cost solution for sensing in the environment. The integration of vision with information from sensors commonly available in a wide range of UAV applications such as a low-cost Inertial Measuring Unit (IMU) and Global Positioning System (GPS) receiver, when integrated properly, provides the ability for an accurately geo-referenced map of the environment, which can be used to locate different types of vegetation and develop effective strategies for environmental management such as weed-spraying.

This paper presents a framework for the integration of sensor information collected from a small, low-flying UAV for constructing geo-referenced maps and for identifying and classifying different types of vegetation within the operating environment. Our approach to mapping seeks to use all of the sensor information from an IMU, GPS and monocular vision camera to construct a joint estimate of the trajectory of the UAV and a dense point feature map of the terrain based on a maximum-likelihood, non-linear least squares approach, building on our previous work (Bryson et al., 2009). The advantage of joint estimation using all of the sensor data together is that the complimentary nature of the different sensors is exploited; vision information allows for more accurate UAV attitude estimates and IMU bias calibration where as the integration of IMU and GPS allows the translation, rotation and scale of the estimated terrain map to be fixed. Our approach to classification is based on generic colour and texture descriptors which can be used to distinguish between different types of vegetation visible in the image data owing to the low-altitude of the UAV flight. A classification algorithm is developed based on supervised training examples of different types of vegetation that are provided to the algorithm by a human expert. The advantage of the approach is that no model information of different types of vegetation is required; the classification algorithm learns the distinguishing features of each vegetation class based on the training examples provided. The classified vegetation is then geo-referenced using the final terrain map and

presented to the user.

In Section 1.1, we discuss related work in environmental monitoring, vision-based mapping and vision-based classification where Section 1.2 provides an overview of the system presented in this paper. Section 2 presents algorithms for the UAV trajectory and terrain map reconstruction. Section 3 presents algorithms for the classification of different vegetation in the map using vision information. Section 4 presents an experimental setup of a small UAV operating over farmland in Queensland, Australia during a weed-finding mission, along with results of both the mapping and classification components of the system. Conclusions and future work are presented in Section 5.

1.1 Related Work

The following subsections discuss related work in the areas of agricultural and aerial mapping, vision-based sensor fusion and classification.

1.1.1 Agricultural Mapping and Remote Sensing

Invasive weeds cause a great deal of damage to agricultural land and native ecosystems each year. If a weed outbreak is not detected and managed quickly, the species can establish a persistent stronghold on the landscape (Lawes and Wallace, 2008). Consequently, it is in the interests of property owners, managers and governments to invest effort into the early detection and ongoing monitoring of invasive weed outbreaks to form informed control strategies such as targeted eradication or containment. The detection of weeds over a landscape presents a challenge in terms of data collection. Ground-based surveys are restricted by terrain accessibility and are poorly suited to monitoring large regions due to the labour and transport costs involved. Consequently, remote sensing is a valuable source of data for weed monitoring, and commonly features in *precision agriculture* management strategies (Zhang et al., 2002). Remotely sensed data from high-flying aerial surveys (Klinken et al., 2007; Bajwa and Tian, 2001; Sandmann and Lertzman, 2003) and satellite remote sensing (Medlin et al., 2000) are typically used for map construction, however these surveys typically provide maps of, at most, only 1-10m resolution which can be insufficient for the detection of certain weeds.

1.1.2 Vision-Based Mapping and Reconstruction

The problem of estimating the 3D structure of a scene using vision sensors is a well studied problem in the structure from motion community (Koch et al., 1998; Pollefeys, 2004) where more popularly, stereo cameras are used to infer the depth in a scene and reconstruct the 3D terrain using multiple stereo image pairs, while simultaneously estimating the pose information of each camera pair. The baseline distance between stereo cameras used to infer scene depth is typically not sufficient in airborne systems (where the ratio of the vehicle altitude above the terrain to the baseline distance is too high); thus in this paper we are interested in the use of monocular camera systems.

The use of camera information alone in this type of 3D terrain reconstruction task has a number of issues principally related to tracking and matching of terrain feature across camera frames and the unobservability in the absolute translation, rotation and scale of the map. There has thus been much interest in the use of other sensors such as a low-cost Inertial Measuring Unit (IMU) for assisting in the mapping process; (Qian et al., 2000) demonstrates the use of vehicle rotation rate information from gyros mounted to the vehicle to assist in the process of matching camera features from frame to frame. Aside from helping to track features, inertial sensors also provide the ability to disambiguate map scale and to some degree the rotation of the map (w.r.t Earth surface coordinates) and with the combination of information from a low-cost Global Positioning System (GPS) receiver, the translation, rotation and scale of the constructed 3D terrain map can be completely fixed.

Aerial photogrammetry is the process of measuring geometric distances and building maps using imagery collected from an airborne vehicle, where often monocular camera systems are used along with other navigation sensors. Historically, ground control points (artificially measured points on the ground which are visible in the imagery) were used to geo-reference images taken to a ground plane to produce maps. The work in (Mostafa and Schwarz, 2000; McCarthy T. and G, 2007) demonstrate more modern approaches to aerial photogrammetry without the need for ground control points where instead navigation data from an on-board IMU and GPS are used to geo-reference imagery taken from an airborne vehicle. Unlike in structure from motion approaches, these works assume that the navigation data is highly accurate (i.e. from high-end IMU and GPS sensors) and thus the map is built by projecting the image data onto the ground using the navigation data without simultaneously correcting the navigation data itself. The accuracy of the final maps are thus limited to the sum of the navigation and imagery errors and no 3D map information is recovered.

(Bryson and Sukkarieh, 2007) and (Pinies et al., 2007) both provide systems for building up a 3D point feature map of the terrain by fusing vision and inertial sensor information in an Extended Kalman Filter (EKF), which simultaneously estimates and corrects for the navigation data. The disadvantage of these approaches is that they can provide only sparse point maps and have issues with map consistency over large areas due to the use of the EKF. In (Clark et al., 2006) the authors use information from poses computed by an on-board IMU-GPS navigation system to assist in a bundle adjustment algorithm for computing dense maps of the environment but don't integrate the IMU information directly into the estimation cycle, and thus still have some issues in the map rotation and scale.

1.1.3 Vision-Based Classification

Species identification in the remote sensing literature is primarily based on spectral reflectance over multiple wavelength channels. Peaks in the visible and near infrared reflectance can be associated with cellular chemical and biological properties of vegetation and are used to discriminate between different species (Hsieh et al., 2001; Yu et al., 2006). Consequently it is common to have data with many spectral channels while achieving only a relatively coarse spatial resolution so that per-pixel maximum likelihood classifiers may be applied. Imagery from the SPOT High Resolution Visible (HRV) or Landsat Thematic Mapper (TM) satellites has been successfully used to study large scale vegetation patterns (over thousands of square

kilometers) and long term temporal patterns across image sequences (Lawes and Wallace, 2008; Guerschman et al., 2003; Robinson and Metternicht, 2005). While low cost, the limited number of channels in the imagery cannot always provide reliable discrimination between similar species of vegetation (Harvey and Hill, 2001; Czaplewski and Patterson, 2004), and the relatively coarse pixel sizes lead to poor performance at detecting low density infestations due to spectrum mixing over the area of the pixels (Klinken et al., 2007). Even for resolutions up to 4m/pixel, one study found that weed infestations of up to 30% coverage over a $200m^2$ area could not be reliably detected due to mixing (Casady et al., 2005). In addition, data from these satellites is not always updated frequently enough for the application of targeted detection and monitoring.

At a higher cost, commercial data is available at high spatial resolutions (1-10m) through airborne multispectral imaging (Lamb and Lamb, 2002; Glenn et al., 2005) or up to 1.65m/pixel with the current highest resolution satellite imaging (Madden, 2009; Casady et al., 2005; Carleer and Wolff, 2004). Most remote sensing and mapping in agriculture uses coarse resolution multispectral imagery, together with independent classifications of each pixel based on radiance in the visible and near infrared spectrum. However, suitable data for this approach can be difficult to obtain, and the resulting classifiers suffer from problems with spectral mixing when the density of a target species is low (Nagendra and Rocchini, 2008). At higher resolutions, where mixing is not a problem, independently classifying pixels leads to a poor signal to noise ratio due to variations in local lighting and local plant appearance (Ehlers et al., 2003; Bajwa and Tian, 2001; Hsieh et al., 2001). Various approaches have accounted for spatial patterns in the data by applying filtering to the output of a classifier (Sun et al., 2003), although more robust performance can be obtained by segmenting neighborhoods of pixels into objects (such as tree crowns) prior to classification (Yu et al., 2006; Culvenor, 2002; Erikson and Olofsson, 2005).

1.2 Overview of the System

Figure 1 illustrates the algorithmic framework and system architecture for the mapping and classification system presented in the paper. The system takes logged sensor data from an IMU, GPS receiver and monocular vision camera collected by a small UAV flying over an area of interest. The goal of the system is to produce a geo-referenced, 3D map of the area, illustrating visual features seen by the camera to their locations in the area and to identify and classify several different types of vegetation, based on their visual appearance. The operation of the system is broken down into two sections.

1.2.1 UAV Trajectory Reconstruction and Map Building

The first section of the system takes IMU, GPS and vision data and produces an accurate 3D map of the terrain, relating features seen in the camera data to their corresponding location and orientation in the world. The system performs the following steps:

1. **INS/GPS Initial Pose Estimation**: IMU and GPS data is used to provide an initial estimate for the position and orientation (pose) history of the UAV during the

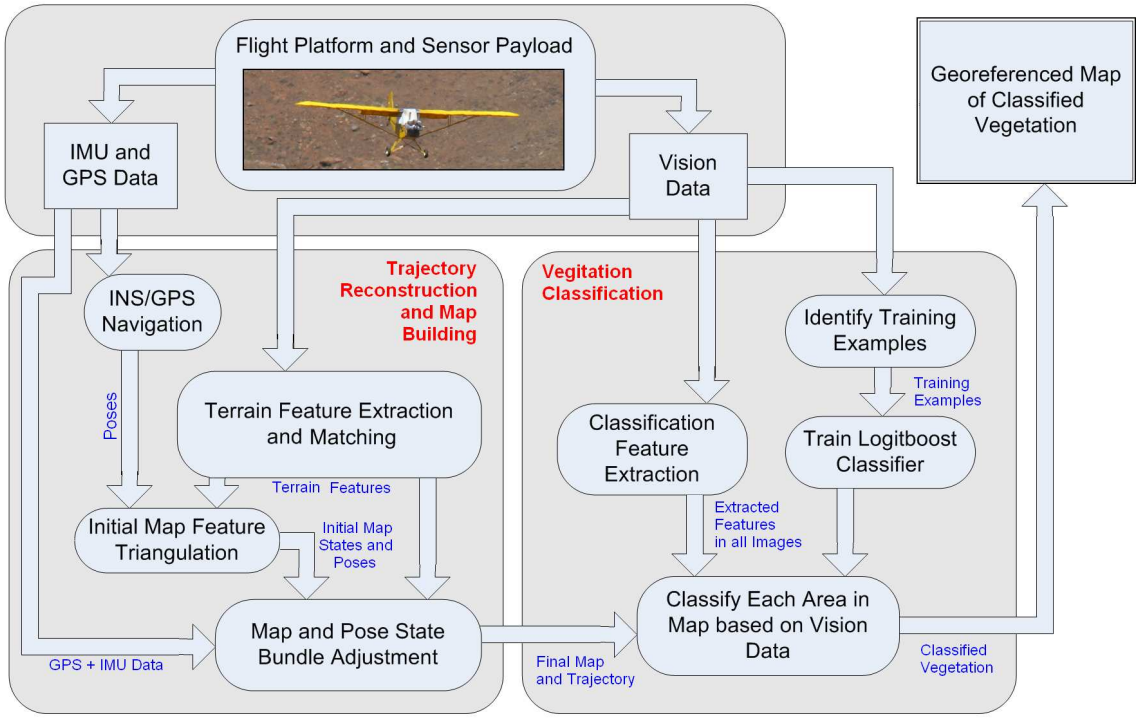

Figure 1: Overview of System Architecture: IMU, GPS and vision data are collected from a sensor payload mounted to a small UAV as it flies over an area of interest. IMU and GPS data are used to initially estimate the position and attitude of the UAV along its trajectory. Point features in the terrain are extracted and matched across frames using the vision data. An initial guess of the 3D position of these point features is made via triangulation, and by using the initial estimated trajectory data. A bundle adjustment procedure is performed over the UAV trajectory and map feature states in order to achieve a final accurate terrain map. For classifying vegetation, examples of different types of vegetation are identified by a human expert in the vision data and used to train a classifier. Patches of the vision data corresponding to each area in the terrain map is then classified into different classes of vegetation.

flight. Data is processed in an Extended Kalman Filter (EKF) architecture. These initial pose estimates are used to help triangulate the 3D position of terrain features and also to provide an initial estimate for the bundle adjustment procedure described below.

2. **Terrain Feature Extraction and Mapping**: Point features in the vision data are extracted using a corner point extractor (Shi and Tomasi, 1994) and tracked across multiple frames using a Lucas-Kanade Tracker (Lucas and Kanade, 1981). Robust outlier rejection in the matches is performed using epipolar constraints and a fundamental matrix calculation between the frames based on a RANSAC method (Torr and Murray, 1997). The resulting feature observations are used to estimate the corresponding 3D location of each feature on the terrain.
3. **Initial Map Feature Triangulation**: Using the matched point features in the vision data and EKF-estimated initial UAV poses, an initial estimate of the 3D positions of

terrain features is made by triangulating feature observations across two observations with the largest baseline. The resulting points are used as an initial estimate for the bundle adjustment procedure described below.

4. **Map and Pose State Bundle Adjustment**: Using all of the sensor data from the IMU, GPS receiver and matched vision features, and starting from the initial vehicle poses and 3D terrain feature positions estimated above, a final estimate for the UAV's trajectory and position of terrain features is made through a non-linear least squares bundle adjustment procedure which accounts for all of the joint relationships between the sensor data. The results of the bundle adjustment procedure is an accurate and consistent, geo-referenced 3D point feature map of the terrain.
5. **Mosaicing and Map Visualisation**: Using the bundle adjusted estimates of the vehicle poses and terrain positions, a photo-mosaic of the terrain is constructed for visualisation purposes by projecting each image taken by the camera onto a ground plane, whose height is taken from the surrounding 3D terrain points.

This section of the system is discussed in more detail in Section 2.

1.2.2 Vegetation Classification

The second section of the system takes vision data and the constructed terrain map data and classifies each section of the terrain into several different classes of vegetation based on visual appearance. The system performs the following steps:

1. **Classification Feature Extraction**: A feature descriptor is chosen based on the colour and texture properties (using a Laplacian pyramid decomposition (Burt and Adelson, 1983; Simoncelli and Freeman, 1995)) of small patches of vision data. The descriptor is applied to the vision data patch associated with each area in the constructed environment map.
2. **Identify Training Examples**: A human weed expert is employed to identify different species of vegetation in a ground based-survey of a small portion of the operating area using a handheld GPS receiver. These examples are then located in both the collected imagery and the constructed environment map, where patches of the image data are used as training features for classification.
3. **Classifier Training**: Based on the training examples of class labelled image patches, the Logitboost algorithm (Friedman et al., 2000) is applied to construct a classifier which maps from the colour and texture descriptors applied to each section of the map to a vegetation class with an associated probability.
4. **Map Classification and Verification**: The classifier is used to determine the class of vegetation associated with each area of the environment map, and the associated class data is projected into the final map visualisation.

This section of the system is discussed in more detail in Section 3.

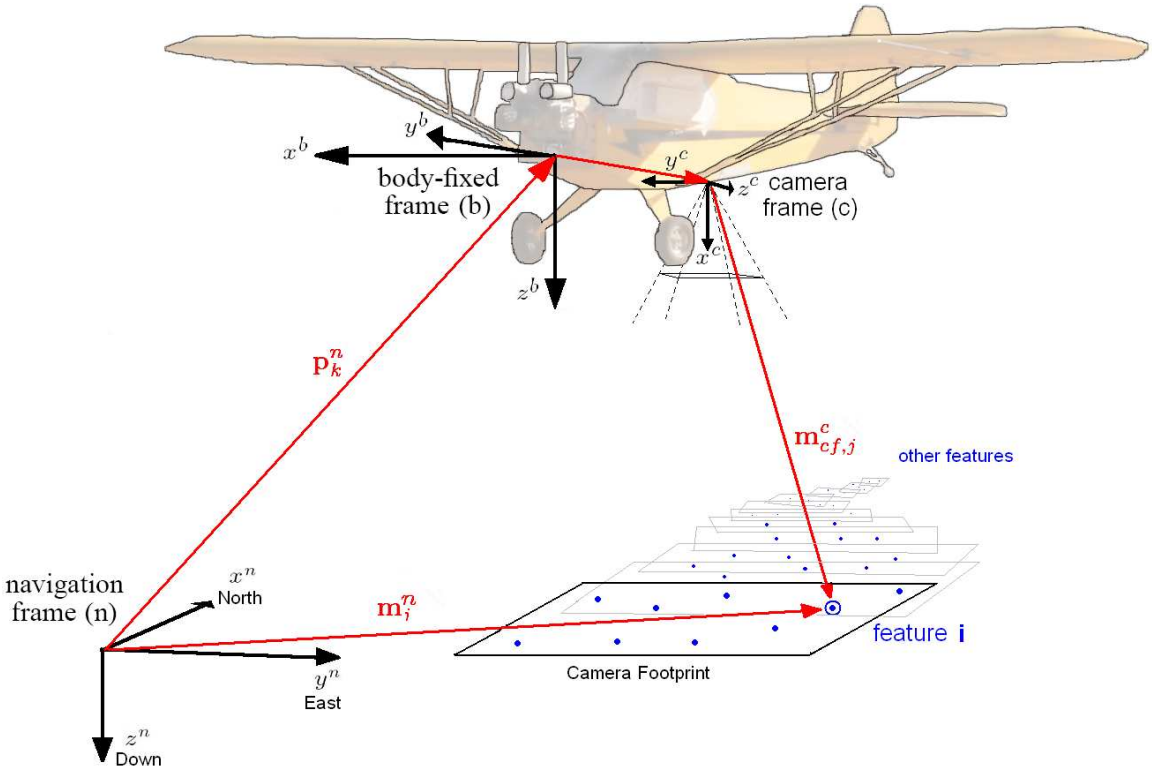

Figure 2: Overview of Frames of Reference, UAV Localisation and Map States during Landmark Observation: Shown are the frames of reference and relationship between the UAV position, map feature position and relative camera observation vector during a camera-based observation of the terrain.

2 UAV Trajectory Reconstruction and Map Building

This section describes our approach to processing sensor information collected from the UAV and the estimation of the vehicle trajectory and 3D structure of the terrain given the data.

2.1 State Vector and Observations

The aim of our trajectory reconstruction and mapping task is to use information from the inertial sensors, GPS and monocular camera to create a dense feature map of the terrain while simultaneously estimating the position and attitude trajectory of the UAV. Our estimated state vector is thus:

$$\begin{aligned}\hat{\mathbf{x}} &= [\mathbf{v}_1^n, \mathbf{p}_1^n, \mathbf{\Psi}_1^n, \mathbf{p}_2^n, \mathbf{\Psi}_2^n, \ldots, \mathbf{p}_K^n, \mathbf{\Psi}_K^n, \ldots \\ &\quad \mathbf{m}_1^n, \mathbf{m}_2^n, \ldots, \mathbf{m}_{N_f}^n, \delta\mathbf{f}^b, \delta\omega_{ib}^b, \delta\mathbf{\Psi}_b^c]^T \end{aligned} \quad (1)$$

This state vector contains three-dimensional vehicle positions $\mathbf{p}_1^n$ to $\mathbf{p}_K^n$ and vehicle three-axis Euler angles $\mathbf{\Psi}_1^n$ to $\mathbf{\Psi}_K^n$, sampled at discrete intervals along the trajectory, where K is the total number of vehicle poses. The superscript n indicates that the positions are referenced w.r.t a terrain-fixed navigation frame located at a set point on the surface of the Earth (the

vehicle's starting location with axes aligned in the North, East and Downwards directions) and the Euler angles represent the rotations required to represent the vehicle body axis w.r.t this frame. $\mathbf{m}_1^n$ to $\mathbf{m}_{N_f}^n$ are the N terrain feature positions, referenced in the terrain-fixed navigation frame n. Additionally, the initial velocity vector $\mathbf{v}_1^n$ of the UAV is also estimated due to it's relationship in the inertial navigation system equations (see Section 2.3.1 for more details).

Along with the vehicle pose and terrain feature states, the trajectory and mapping algorithm also estimates $\delta\mathbf{f}^b$ and $\delta\omega_{ib}^b$, constant IMU biases, three for the accelerometers and three for the gyros respectively, and $\delta\mathbf{\Psi}_b^c$, the misalignment angles of the camera orientation w.r.t the IMU. Particularly in the case of IMU biases, which change each time the IMU is powered on (switch on-off biases), these parameters are generally difficult to calibrate for on the ground before flight.

The sensor observations from the IMU, GPS receiver and monocular camera, used to help estimate the state $\hat{\mathbf{x}}$, are grouped into the observation vector:

$$\begin{aligned}\mathbf{z} &= [\hat{\mathbf{f}}_2^b, \hat{\omega}_{ib,2}^b, \ldots, \hat{\mathbf{f}}_K^b, \hat{\omega}_{ib,K}^b, \ldots \\ &\quad \mathbf{p}_{GPS,1}^n, \ldots, \mathbf{p}_{GPS,N_g}^n, \mathbf{v}_{GPS,1}^n, \ldots, \mathbf{v}_{GPS,N_g}^n, u_1, v_1, \ldots, u_{N_c}, v_{N_c}]^T \end{aligned} \quad (2)$$

where $\hat{\mathbf{f}}_2^b$ to $\hat{\mathbf{f}}_K^b$ are the $K-1$ three-axis IMU accelerometer readings, $\hat{\omega}_{ib,2}^b$ to $\hat{\omega}_{ib,K}^b$ are the $K-1$ three-axis IMU gyro readings (both referenced w.r.t a vehicle body-fixed frame b). As a matter of notation, only IMU measurements taken at the time of the second pose $[\mathbf{p}_2^n, \mathbf{\Psi}_2^n]$ and onwards are used. $\mathbf{p}_{GPS,1}^n$ to $\mathbf{p}_{GPS,N_g}^n$ and $\mathbf{v}_{GPS,1}^n$ to $\mathbf{v}_{GPS,N_g}^n$ are the N_g GPS position and velocity measurements respectively, referenced in the n frame, and u_1, v_1 to u_{N_c}, v_{N_c} are the N_c image pixel coordinates of feature observations made by the camera. The relationship between the different frames of reference and UAV trajectory and map states is illustrated in Figure 2.

Observations from the IMU and GPS receiver are taken as the raw measurements that come directly from these sensors, whereas the matched image pixel coordinate measurements of terrain features are generated from the monocular vision data during a feature extraction and matching phase described below.

2.2 Vision Feature Extraction and Matching

An on-board monocular camera is used to take images of the terrain below the aircraft from which observations of point-features in the terrain are made. In this subsection we describe the feature extraction and matching process, in which point feature observations in the camera frame are made and related to terrain features in the environment. The feature extraction and matching process performs the following steps:

1. **Offline Camera Calibration**: Before flight, the intrinsic parameters and lens distortion parameters for the camera system are computed in an off-line procedure (Bouguet, 2009). The lens distortion parameters are applied to undistort each frame from the camera before any other processing occurs.

2. **Initial Feature Extraction**: Point features in the vision data are extracted in the first vision frame using a corner point extractor (Shi and Tomasi, 1994) which finds pixels in the image with large eigenvalues in their associated local intensity gradient matricies.
3. **Tracking Features Across Frames**: After features have been located in the first image, the algorithm moves forward through each subsequent frame and attempts to track the location in the new frame of good features found in the last frame. For each point feature located in the previous frame, the corresponding position of the feature in the next vision frame (assuming at least some overlap in the imagery) is computed using a Lucas-Kanade (LK) Tracker (Lucas and Kanade, 1981) which tracks the feature using local intensity gradient information.
4. **Robust Outlier Rejection using Epipolar Constraints**: The result of the LK tracker is a set of potential tracks of the features from the previous frame, but containing a significant number of outliers and bad matches. These outliers are detected and rejected by using the set of matches to robustly calculate the fundamental matrix between the frames using a RANSAC method (Torr and Murray, 1997). The fundamental matrix defines the epipolar geometry between the two frames; those feature matches found to violate the epipolar constraint (within a margin of error defined in the images) are rejected. A record of the same feature matched over multiple frames is kept when such occurs, and thus a large number of features are tracked over more than two frames while the feature remains in the field of view of the camera.
5. **New Feature Extraction**: After a list of good feature tracks has been found in the current frame, more point features are extracted using the corner point extractor (Shi and Tomasi, 1994), but where the search in the image is limited to areas where no feature tracks have been found. This provides a way of extracting new features as they come into the field of view of the camera, in the parts of the image that contains no overlap with previous frames.

The process assumes that there is a reasonable amount of overlap in the images (i.e. that the movement of features from one frame to the next is small w.r.t the whole image frame). In practice, the approaches works well for our experimental setup described in Section 4.1 where apparent feature movement from one frame to the next varies from about 100 to 300 pixels in a 1024x768 pixel image. To this end features also stay within the field of view for approximately 3-7 frames and where tracking performs correctly, each observation will be associated with a single terrain feature. An example of the process in our motivating application is presented in Section 4.2.1. The resulting list of feature locations in each image frame along with a record of which features correspond to each other and thus to the same point in the terrain is kept and used in the trajectory and mapping estimation as described in the sections below.

2.3 Sensor Models for Bundle Adjustment

Our approach to estimation of the state vector described in Equation 1 using the observations from the IMU, GPS and monocular vision camera encapsulated in Equation 2 is based on maximum-likelihood; that is we want to estimate the optimal state vector $\hat{\mathbf{x}}$ which most

probably generated the sensor measurements recorded, subject to the assumed levels of error and noise in each sensor. In order to perform this estimation task, we firstly require a set of non-linear relationships that describe the expected sensor observations to be made given a particular value of the states/parameters, otherwise known as sensor model equations. The following subsections describe these relationships for the IMU, GPS receiver and monocular camera point feature observations.

2.3.1 Inertial Sensor Model

The on-board IMU provides high-frequency measurements of the body-fixed specific force and rotation rates of the vehicle which can be used to dead-reckon the position, velocity and attitude of the vehicle forward in time using the following equations:

$$\mathbf{p}_k^n = \mathbf{p}_{k-1}^n + \mathbf{v}_k^n \Delta t \quad (3)$$

$$\mathbf{v}_k^n = \mathbf{v}_{k-1}^n + [\mathbf{C}_b^n(\hat{\mathbf{f}}^b - \delta\mathbf{f}^b) + \mathbf{g}^n]\Delta t \quad (4)$$

$$\mathbf{\Psi}_k^n = \mathbf{\Psi}_{k-1}^n + [\mathbf{E}_b^n(\hat{\omega}_{ib}^b - \delta\omega_{ib}^b)]\Delta t \quad (5)$$

where $\mathbf{v}^n$ is the terrain-fixed navigation frame vehicle velocity, Δt is the time difference between the k and $k-1$ discrete time segments and $\mathbf{g}^n = [0, 0, g]^T$ is the vector of acceleration due to gravity in the local navigation frame ($g = 9.81m/s^2$). $\mathbf{E}_b^n$ is the body to navigation frame rotation rate transformation matrix and $\mathbf{C}_b^n$ is the Direction Cosine Matrix (DCM) transformation from the body to the terrain-fixed navigation frame, both matricies being a function of the Euler angles. The inertial navigation equations in the vehicle process model states are simplified so as to treat the local navigation frame as an inertial frame of reference by ignoring the small coriolis and centripetal accelerations and rotation rate which are incurred by the Earth's rotation (i.e. an inertial frame mechanisation (Titterton and Weston, 1997)). This approach is validated by the fact that the vehicle operates over a small geographic area of the Earth's surface (with respect to the curvature of the earth) and for a short amount of time (relative to the rotational period of the earth). The vectors $\hat{\mathbf{f}}^b$ and $\hat{\omega}_{ib}^b$ are the accelerometer specific force vector reading and gyroscope rotation rate reading respectively, and $\delta\mathbf{f}^b$ and $\delta\omega_{ib}^b$ are the accelerometer and gyro biases respectively.

Equations 3 to 5 can be rearranged to represent the IMU observations as a function of the pose states:

$$\hat{\mathbf{f}}_k^b = \mathbf{C}_n^b \frac{1}{\Delta t^2}[\mathbf{p}_k^n - 2\mathbf{p}_{k-1}^n + \mathbf{p}_{k-2}^n + \Delta t^2 \mathbf{g}^n] + \delta\mathbf{f}^b \quad (6)$$

$$\hat{\omega}_{ib,k}^b = (\mathbf{E}_b^n)^T \frac{1}{\Delta t}[\mathbf{\Psi}_k^n - \mathbf{\Psi}_{k-1}^n] + \delta\omega_{ib}^b \quad (7)$$

where $\mathbf{C}_n^b = (\mathbf{C}_b^n)^T$ is the DCM transformation from the n frame to the b frame. Note that in this form we have eliminated the velocity term by substituting Equation 3 into Equation 4. In this form each measurement from the IMU can be thought of as connecting subsequent vehicle poses together, where each measurement from the gyros applies a noisy constraint over two subsequent sets of vehicle Euler angles and each accelerometer measurement applies a noisy constraint over one set of Euler angles and three subsequent vehicle positions.

Since accelerometer observations are related to three subsequent poses (rather than two as in the gyro case), the representation requires the estimation of two more positions than the

number of accelerometer measurements, and thus position state $\mathbf{p}_{k-2}^n$ in the case that $k = 2$ is unobservable in most cases. In order to overcome this, the first accelerometer measurement $\hat{\mathbf{f}}_2^b$ is related to the pose state via the alternative version of the IMU sensor model equation:

$$\hat{\mathbf{f}}_2^b = \mathbf{C}_n^b \frac{1}{\Delta t^2}[\mathbf{p}_2^n - \mathbf{p}_1^n + \Delta t \mathbf{v}_1^n + \Delta t^2 \mathbf{g}^n] + \delta \mathbf{f}^b \tag{8}$$

which relates the measurement to the first two position states and the initial velocity of the UAV.

2.3.2 GPS Sensor Models

A GPS receiver on-board the vehicle makes observations of the vehicle position and velocity which are subsequently referenced to the terrain-fixed navigation frame n. The position observations are thus related to the estimated vehicle position states via the GPS position observation model equation:

$$\mathbf{p}_{GPS,k}^n = \mathbf{p}_k^n + \mathbf{C}_b^n \mathbf{l}_{GPS}^b \tag{9}$$

where $\mathbf{p}_k^n$ is the position of the vehicle at time segment k (when the GPS observation is made) and $\mathbf{l}_{GPS}^b$ is the lever-arm of the GPS antenna w.r.t the IMU.

Assuming the time between two estimated position states is small, the GPS velocity measurement can be related to two subsequent estimated vehicle position states via the GPS velocity observation model equation:

$$\mathbf{v}_{GPS,k}^n = \frac{1}{\Delta t}[\mathbf{p}_k^n - \mathbf{p}_{k-1}^n] \tag{10}$$

where Δt is the time difference between k and $k-1$ and the velocity induced by the GPS lever-arm due to UAV rotation rate is negligible and assumed zero.

2.3.3 Monocular Camera Sensor Models

Assuming pixel referenced observations of terrain features have been extracted and matched to a given terrain feature (as described in Section 2.2), the jth feature observation is made up of the u_j and v_j pixel coordinates in the image which are related to the estimated state vector $\hat{\mathbf{x}}$ via the pinhole camera model observation equation:

$$u_j = \hat{f}_u(\frac{y_j^c}{x_j^c}) + \hat{u}_0 \tag{11}$$

$$v_j = \hat{f}_v(\frac{z_j^c}{x_j^c}) + \hat{v}_0 \tag{12}$$

where x_j^c, y_j^c and z_j^c are the cartesian coordinates of $\mathbf{m}_{cf,j}^c$ the feature position w.r.t the camera, measured in the frame c, fixed to the monocular camera. $\hat{f}_u$, $\hat{f}_v$, $\hat{u}_0$ and $\hat{v}_0$ are the pinhole camera model parameters (the horizontal and vertical focal lengths and principle point) of the monocular camera. These parameters are calculated in an offline calibration procedure (Bouguet, 2009).

The feature position vector w.r.t the camera, $\mathbf{m}_{cf,j}^c$, is related to the other state vector states via:

$$\mathbf{m}_{cf,j}^c = (\mathbf{I} + [\times \delta \mathbf{\Psi}_b^c])\hat{\mathbf{C}}_b^c \mathbf{C}_n^b [\mathbf{m}_i^n - \mathbf{p}_k^n - \mathbf{C}_b^n \mathbf{l}_{cam}^b] \tag{13}$$

where the jth observation is found to correspond to the ith feature ($\mathbf{m}_i^n$). $\hat{\mathbf{C}}_b^c$ is the initially guessed DCM transformation from the body-fixed frame b to the camera-fixed frame c. This value is guessed from the mechanical mounting of the camera to the UAV and is expected to be accurate to approximately $\pm 10^o$ of rotation. $[\times \delta\mathbf{\Psi}_b^c]$ is the skew symmetric matrix of $\delta\mathbf{\Psi}_b^c$, the misalignment angles of the camera w.r.t the body frame. The relationship between the actual ($\mathbf{C}_b^c$), guessed and misalignment angles of the body to camera frame rotation is given via:

$$\mathbf{C}_b^c = [\mathbf{I}_{3x3} + [\times \delta\mathbf{\Psi}_b^c]]\hat{\mathbf{C}}_b^c \tag{14}$$

$\mathbf{\Psi}_b^c$ forms part of the estimated state vector $\hat{\mathbf{x}}$ as described in Section 2.1.

$\mathbf{l}_{cam}^b$ is the lever-arm of the camera w.r.t the IMU. It should be noted that we have chosen to estimate the errors in the camera angular alignment but not the lever-arm, as the size of the lever-arm relative to the range to features on the ground is small. Any errors in the measured lever-arm have an insignificant effect on the mapping accuracy, whereas errors in the camera angular alignment can cause significant mapping errors.

2.4 Initial Estimates for Bundle Adjustment

In order to produce an maximum-likelihood estimate of the state vector $\hat{\mathbf{x}}$, an iterative bundle adjustment method is applied as described in Section 2.5 below. This procedure requires a rough initial guess of the state vector $\hat{\mathbf{x}}$ in order to converge towards the correct solution. The following subsections describe the methods for generating a initial estimate of the UAV trajectory and mapping state vector presented in Section 2.1. These methods provide a quick and rough estimate of the initial state which does not account for all of the sensor observations and does not account for the joint estimation of UAV trajectory and mapping states; the aim is thus simply to provide a starting point for the bundle adjustment algorithm which will go on to provide the final result.

2.4.1 Initial Trajectory Estimates and Sensor Parameters

The initial estimates for the vehicle position ($\mathbf{p}_1^n$ to $\mathbf{p}_K^n$) and attitude ($\mathbf{\Psi}_1^n$ to $\mathbf{\Psi}_K^n$) states are made by applying an EKF to sequentially estimate these states using the IMU and GPS data. The initial position, velocity and attitude of the UAV while on the ground and stationary before takeoff is used as an initial state for the EKF, and is computed via the GPS data, accelerometer/tilt data and an external heading measurement (made by a hand-held compass on the ground). An EKF prediction model, is used to predict forward the position, velocity and attitude estimate from timestep k to $k+1$ using the inertial navigation mechanisation in Equations 3 to 5. At the time of GPS position and velocity observations, the estimates are corrected in an EKF update stage using the sensor models in Equations 9 and 10. For more details on how this technique is applied, the reader is referred to (Kim et al., 2003; Sukkarieh et al., 2003) but the details are omitted here due to space constraints.

In addition to the initial UAV trajectory states, initial guesses for the IMU biases ($\delta\mathbf{f}^b$ and $\delta\omega_{ib}^b$) and camera misalignment $\delta\mathbf{\Psi}_b^c$ are all set equal to zero.

2.4.2 Initial Feature Triangulation

In order to provide an initial guess of the positions of each point feature in the map, an initial feature triangulation routine is performed using the associated feature observations and the initial poses estimated by the IMU-GPS EKF procedure discussed above. For each set of tracked pixel features that correspond to a single map location, we group together all camera pixel observations. For each of these observations we calculate the unit vector $\bar{\mathbf{u}}_j^n$ of the feature observation direction and the position from which the observation was made, w.r.t the local terrain navigation frame n:

$$\bar{\mathbf{u}}_j^n = \mathbf{C}_b^n \hat{\mathbf{C}}_c^b \bar{\mathbf{m}}_{cf,j}^c \tag{15}$$

$$\mathbf{y}_j^n = \mathbf{p}_k^n + \mathbf{C}_b^n \mathbf{l}_{cam}^b \tag{16}$$

where $\mathbf{p}_k^n$ and $\mathbf{C}_b^n$ are taken from the IMU-GPS EKF solution in Section 2.4.1 and $\bar{\mathbf{m}}_{cf,j}^c$ is the unit vector of the feature position w.r.t the camera measured in the camera frame, which is calculated using the pixel coordinate of the observation:

$$\bar{\mathbf{m}}_{cf,j}^c = [\frac{1}{r}, \frac{u_j - \hat{u}_0}{r\hat{f}_u}, \frac{v_j - \hat{v}_0}{r\hat{f}_v}]^T \tag{17}$$

$$r = 1 + (\frac{u_j - \hat{u}_0}{\hat{f}_u})^2 + (\frac{v_j - \hat{v}_0}{\hat{f}_v})^2 \tag{18}$$

By calculating the dot product between the every combination of two unit vectors, we find the two unit vectors with the greatest angular separation (i.e. largest dot product) and use these two measurements to calculate the feature position $\mathbf{m}_{i,init}^n$ as the closest point between the two feature observation lines-of-sight:

$$\mathbf{m}_{i,init}^n = \frac{1}{2}(\mathbf{y}_1^n + \mathbf{y}_2^n + p_1.\bar{\mathbf{u}}_1^n + p_2.\bar{\mathbf{u}}_2^n) \tag{19}$$

$$p_1 = \frac{((\mathbf{y}_2^n - \mathbf{y}_1^n) \times \bar{\mathbf{u}}_2^n) \cdot (\bar{\mathbf{u}}_1^n \times \bar{\mathbf{u}}_2^n)}{|\bar{\mathbf{u}}_1^n \times \bar{\mathbf{u}}_2^n|^2} \tag{20}$$

$$p_2 = \frac{((\mathbf{y}_1^n - \mathbf{y}_2^n) \times \bar{\mathbf{u}}_1^n) \cdot (\bar{\mathbf{u}}_2^n \times \bar{\mathbf{u}}_1^n)}{|\bar{\mathbf{u}}_2^n \times \bar{\mathbf{u}}_1^n|^2} \tag{21}$$

If the minimum distance between the two line of sights is above a given threshold, an error is assumed to have occurred in the matching process and the feature information is rejected before bundle adjustment. Otherwise, the resulting guess of the initial feature position $\mathbf{m}_{i,init}^n$ is then added to the initial state vector $\hat{\mathbf{x}}$ and refined during the estimation procedure.

2.5 Non-Linear Least Squares and Bundle Adjustment Procedure

The aim of our UAV trajectory and map reconstruction algorithm is, using the observation vector $\mathbf{z}$, to estimate the state vector $\hat{\mathbf{x}}$ which minimises the weighted Sum of Squared Error (SSE) cost function:

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}))^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})) \tag{22}$$

where $\mathbf{R}$ is the covariance of the expected noise on the observations contained within the observation vector $\mathbf{z}$ and $\mathbf{h}(\hat{\mathbf{x}})$ is the predicted value of our observation vector given a parameter estimate $\hat{\mathbf{x}}$. The function $\mathbf{h}(.)$ is composed of the set of nonlinear functions that

relate the estimated parameter values to the different sensor observations made as described in Section 2.3.

This type of problem formulation (generally referred to as non-linear least squares) is typically solved by a numeric optimisation method which chooses an initial state estimate and iteratively moves towards the state vector that minimises the cost function in Equation 22 by making linear approximations of the non-linear sensor model $\mathbf{h}(.)$. Commonly used optimisation methods include Newton's method, the Gauss-Newton approximation (Nocedal and Wright, 2006) and the Levenberg-Marquardt algorithm (which is commonly used in vision-based bundle adjustment (Triggs et al., 2000)). In our approach we use a Gauss-Newton approximation (by using the information matrix and information vector) due to its relatively fast computation time and convergence properties.

Moving on from an initial state estimate, at each iteration, the Jacobian matrix of the composed sensor model function $\mathbf{h}(.)$ is evaluated using the last state estimate $\hat{\mathbf{x}}$:

$$\nabla \mathbf{H} = \left(\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right)_{\mathbf{x}=\hat{\mathbf{x}}} \tag{23}$$

The jacobian matrix is then used to compute the information matrix $\mathbf{Y}$ and information vector $\mathbf{y}$:

$$\mathbf{Y} = \nabla \mathbf{H}^T \mathbf{R}^{-1} \nabla \mathbf{H} \tag{24}$$

$$\mathbf{y} = \nabla \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})) \tag{25}$$

from which the estimate is updated:

$$\hat{\mathbf{x}} = \hat{\mathbf{x}} + \delta \hat{\mathbf{x}} \tag{26}$$

where $\delta \hat{\mathbf{x}}$ is the solution to the linear system:

$$\mathbf{y} = \mathbf{Y} \delta \hat{\mathbf{x}} \tag{27}$$

The iteration is performed until the euclidian norm of the vector $\delta \hat{\mathbf{x}}$ falls below a specified threshold, at which point the algorithm has converged on a state estimate.

The bundle adjustment procedure thus performs the following steps:

1. **Initial Sensor Processing and Initial State Estimates**: The total observation vector $\mathbf{z}$ (Equation 2) is constructed using the collected sensor information and extracted and matched vision features. The initial estimate for the total state vector $\hat{\mathbf{x}}$ (Equation 1) is computed based on the sensor data as described in Section 2.4.
2. **Construction of the Predicted Observation Vector $\mathbf{h}(\hat{\mathbf{x}})$ and Jacobian Matrix $\nabla \mathbf{H}$**: Given the latest state vector estimate, at each iteration, the observation Jacobian matrix $\nabla \mathbf{H}$ is evaluated by computing the Jacobians of the non-linear functions for the IMU data in Equations 6 and 7, GPS data in Equations 9 and 10 and camera data in Equations 11 and 12 w.r.t the estimated state vector $\hat{\mathbf{x}}$. Additionally to the calculation of $\nabla \mathbf{H}$, the sensor model Equations 6, 7, 9, 10, 11, 12 and 13 are used to compute the predicted vector of observations (i.e. $\mathbf{h}(\hat{\mathbf{x}})$) from the current iteration of our estimated state vector $\hat{\mathbf{x}}$. The jacobian matrix $\nabla \mathbf{H}$ is very large (based on the large dimensions of the vectors $\hat{\mathbf{x}}$ and $\mathbf{z}$), but has a large degree of sparsity;

sparse matrix methods are thus used to store $\nabla\mathbf{H}$ as a sparse, column-compressed matrix.

3. **Non-linear Least Squares Solving**: Once $\nabla\mathbf{H}$ and $\mathbf{h}(\hat{\mathbf{x}})$ have been computed for the current iteration of the estimator, the information matrix and information vector are calculated using Equations 24 and 25. The state update vector $\delta\hat{\mathbf{x}}$ is then computed through the solution to Equation 27 which is solved using sparse matrix techniques. The rows and columns of the information matrix and vector are permuted using an approximate minimum degree ordering (Amestoy et al., 1996). Sparse Cholesky decomposition (Anderson et al., 1999) is performed on the permuted matrix term and eventually used to solve for $\delta\hat{\mathbf{x}}$. A new estimate for the state vector $\hat{\mathbf{x}}$ is then computed using Equation 26. The norm of the state update vector ($|\delta\hat{\mathbf{x}}|$) is then computed; if this value is smaller than a specified tolerance value, the bundle adjustment scheme has converged to a solution and the procedure is finished, otherwise further iterations of the algorithm are performed by returning to step 2 with the updated state estimate.

2.6 Terrain Mosaicing and Visualisation

In order for the constructed map information to be useful for potential human-users of the system, a visualisation of the data is created through the construction of a photo-mosaic which is built by projecting each image of the terrain captured by the camera on to a 2D map of the environment based on the UAV trajectory information and the terrain height information provided by the terrain feature map.

Given the calibrated intrinsic parameters of the camera, one can compute the camera frame unit vector directions of the pixels corresponding to the four corners of the captured image (defined by $[\bar{\mathbf{m}}^c_1, \bar{\mathbf{m}}^c_2, \bar{\mathbf{m}}^c_3, \bar{\mathbf{m}}^c_4]$ where the subscript defines the corner number) using Equations 17 and 18. In this case, the pixel coordinates u_j and v_j for the four corners are given by

$$\begin{aligned} corner_1 &\rightarrow u_j = 0, v_j = 0 \\ corner_2 &\rightarrow u_j = W-1, v_j = 0 \\ corner_3 &\rightarrow u_j = 0, v_j = H-1 \\ corner_4 &\rightarrow u_j = W-1, v_j = H-1 \end{aligned}$$

where W is the width of the image in pixels and H is the height of the image in pixels. Given the position and orientation of the UAV at the time of image capture, these unit vector directions can be transformed from camera-frame to navigation-frame coordinates (i.e. $[\bar{\mathbf{m}}^n_{1,j}, \bar{\mathbf{m}}^n_{2,j}, \bar{\mathbf{m}}^n_{3,j}, \bar{\mathbf{m}}^n_{4,j}]$) using Equation 15. Given the expected downwards position coordinate of the terrain directly below the UAV, $p_{z,ground}$, and the downwards component of the position of the UAV, p_z, (both available from the bundle adjusted estimate), the distance from the ground plane to the UAV h is computed as:

$$h = p_{z,ground} - p_z \tag{28}$$

In this case, the ground height $p_{z,ground}$ is computed by averaging the height of all mapped terrain features within a given area of the terrain directly below the vehicle, in order to arrive at a single value. Given h, the north $(m^n_{x,1,j}, m^n_{x,2,j}, m^n_{x,3,j}, m^n_{x,4,j})$ and east

$(m^n_{y,1,j}, m^n_{y,2,j}, m^n_{y,3,j}, m^n_{y,4,j})$ components of the ground locations corresponding the pixels in each of the four corners of the image can be calculated for each corner via:

$$\begin{bmatrix} m^n_{x,k,j} \\ m^n_{y,k,j} \end{bmatrix} = \begin{bmatrix} \frac{h}{\bar{\mathbf{m}}^n_{k,j,z}} \bar{\mathbf{m}}^n_{k,j,x} + \mathbf{y}^n_{j,x} \\ \frac{h}{\bar{\mathbf{m}}^n_{k,j,z}} \bar{\mathbf{m}}^n_{k,j,y} + \mathbf{y}^n_{j,y} \end{bmatrix} \tag{29}$$

where the subscripts x, y, z are used to indicate the north, east and down components of vectors in the navigation frame, the subscript k is used to indicate the corners, 1 to 4, and $\mathbf{y}^n_j$ is the position of the centre of the camera lens when the image was taken, computed using Equation 16. Once the ground position of the pixels in the four corners of the image are known, the remaining pixels in the image can also be projected on the ground plane through 2D linear interpolation between these points, and thus the entire image is projected onto the ground plane.

This procedure is repeated for each of the camera frames, until all frames are projected onto a single mosaic. The method accounts for variations in the ground height over the distance of the map, however it assumes that the ground plane is relatively flat across the distance of the camera footprint, and thus can cause some misalignment between frames when the ground height varies sharply (i.e. cliffs or step hills). In applications where variations in the terrain are great, complete 3D texturing of the terrain feature map (Johnson-Roberson et al., 2009), using the 3D point feature map built in our approach, is an alternative option for map visualisation. It was found in our experimental application that the ground was relatively flat, thus producing visually consistent mosaic maps in most cases.

3 Vegetation Classification

We have implemented a machine learning vision approach to classify plant species in the UAV imagery based on colour and texture descriptors. An vision-spectrum approach is unconventional for agricultural mapping because the level of detail needed for reliable visual identification of plant species is not usually available from satellite and manned aircraft remote sensing. In our case it is made feasible by the spatial resolutions of up to 4cm/pixel acquired by the low-flying UAV, at altitudes of approximately 100m from the ground.

Visual identification of natural objects is a challenging problem due to variations in illumination, viewing angle and even in appearance between individuals of the same class. Shape and texture properties have been used successfully for classification of weeds in closely managed environments such as crop fields or orchards (Lamb and Lamb, 2002; Bajwa and Tian, 2001; Ye et al., 2007; Alchanatis et al., 2005), but in this problem we are dealing with irregularly distributed species over a natural environment.

We formulate the classification and post processing steps to generate sparse point labels of tree crowns. This compact representation is desired for overlaying labels onto the reconstructed imagery, and to enable further spatial modelling of this data as a future endeavor.

3.1 Quantitative Feature Descriptors

Instead of using the raw pixels of an image directly as inputs to a classification algorithm, it is more robust to extract a set of visual features that quantify appearance properties such as colour and texture in an image.

There have been a large number of texture descriptors proposed in the fields of computer vision and pattern recognition, many of which have been applied to plant species classification from digital photographs. Statistics of co-occurrence matrices (such as energy, homogeneity and inertia) have demonstrated promising results in foliage classification (Yu et al., 2006; Samal et al., 2006), but this type of approach has fallen out of popularity in favour of descriptions that naturally extend to multiple scales and orientations. Gabor filters are a class of quasi-periodic, oriented filters that can be convolved with an image to obtain responses analogous to the processing that occurs in a human visual cortex. Banks of Gabor filters have been coupled with machine learning algorithms such as support vector machines to classify vegetation (Tang et al., 1999; Tang et al., 2003; Manjunath and Madden, 1996). Alternatively, pyramid decompositions or frequency space discrete wavelet transforms may be applied to decompose an image pattern(Chang and Kuo, 1993; Li and Shawe-Taylor, 2005).

Because the aircraft's viewing angle and altitude remain relatively stable, we primarily require a description of appearance that is invariant to translation. We also desire it to be insensitive to lighting angle and small changes in perspective.

From the large choice of texture measures available, we have applied a feature extraction procedure based on a colour space transform, Laplacian pyramid decomposition (Burt and Adelson, 1983; Simoncelli and Freeman, 1995) and block statistics. As a first stage of processing, the red-green-blue colour imagery is transformed into a new colour-space to better indicate shading and colour. This was originally intended to be a Hue-Saturation-Value representation, however it was found that hue varied little over different types of vegetation, but varied greatly in regions of low saturation, where it is highly sensitive to the actual pixel values. Due to these problems with the behavior of the hue component we instead use a Luminance-Chrominance Red-Chrominance Blue (YCrCb) colour-space which describes the colour-space using luminance (Y) and red (Cr) and blue (Cb) chrominance.

Following this transformation, a Laplacian pyramid implementation was used to decompose the image into multiple channels of increasingly coarse detail (Heeger and Bergen, 1995; Burt and Adelson, 1983). To construct a Laplacian image pyramid efficiently, the input image is repeatedly smoothed with a Gaussian convolution filter and differenced from the original to obtain a layer of detail. The smoothed image is then sub-sampled (usually by a factor of 2 in each dimension) and the procedure repeated recursively. This subtraction of a Gaussian blurred image is approximately equivalent to convolving with the Laplacian of a Gaussian filter. Given that the image data is collected at a constant altitude, we do not need to deal with changes of input resolution.

Because the dimensions of the laplacian pyramid decrease by a factor of two for each layer, the resulting outputs are at different resolutions. In this implementation, the recursive

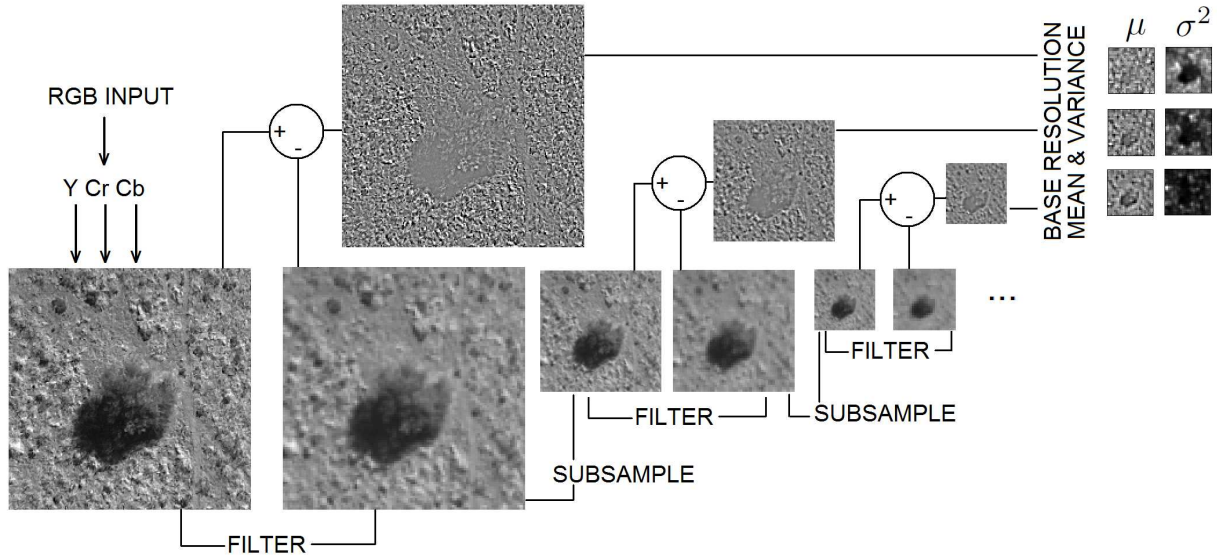

Figure 3: Extracting Texture Descriptors from a Digital Image (luminance channel depicted): The Laplacian pyramid decomposition provides outputs at multiple resolutions, that are then reduced through blocks statistics to a base resolution that is used with a machine learning algorithm for classification.

pyramid decomposition was applied 5 times so that a value in the coarsest layer corresponds to a 16×16 block of pixels in the original image. We reduce all the higher resolution layers of the pyramid to this base resolution by taking the mean and variance of their responses over blocks of equivalent area. Thus in this example we have mean and variance values over the first 4 layers, and the pyramid values of the last. This yields 9 dimensions per channel, over three channels giving 27 features at one sixteenth the resolution. A block diagram of this feature extraction pipeline is shown in Figure 3.

Selection of the number of pyramid layers was a trade between the number of feature dimensions (beneficial to classification), and the resolution of the classifier output, which must be enough to resolve tree crowns. For our 4cm/pixel imagery, it was found that a 5 level pyramid was a good compromise, corresponding to a physical region of approximately 64×64cm per classification output.

3.2 Classification with Logitboost

Once a set of feature dimensions have been extracted from the imagery, it is necessary to use a classification algorithm to map the appearance in texture space into a class membership probability. In the dataset obtained in our experimental setup described in Section 4.1, consultation with weed experts confirmed that there were four primary species of vegetation present in the imagery: Eucalyptus Trees (Eucalyptus coolabah), Prickly Acacia (Acacia nilotica), Parkinsonia (Parkinsonia aculeata), and Mitchell grass (various species). Two of these (Prickly Acacia and Parkinsonia) are formally identified as weeds of national significance in Australia, while the Eucalypt is a native species of interest. Defining the classification problem, these three types of tree were each assigned a class, and a fourth null

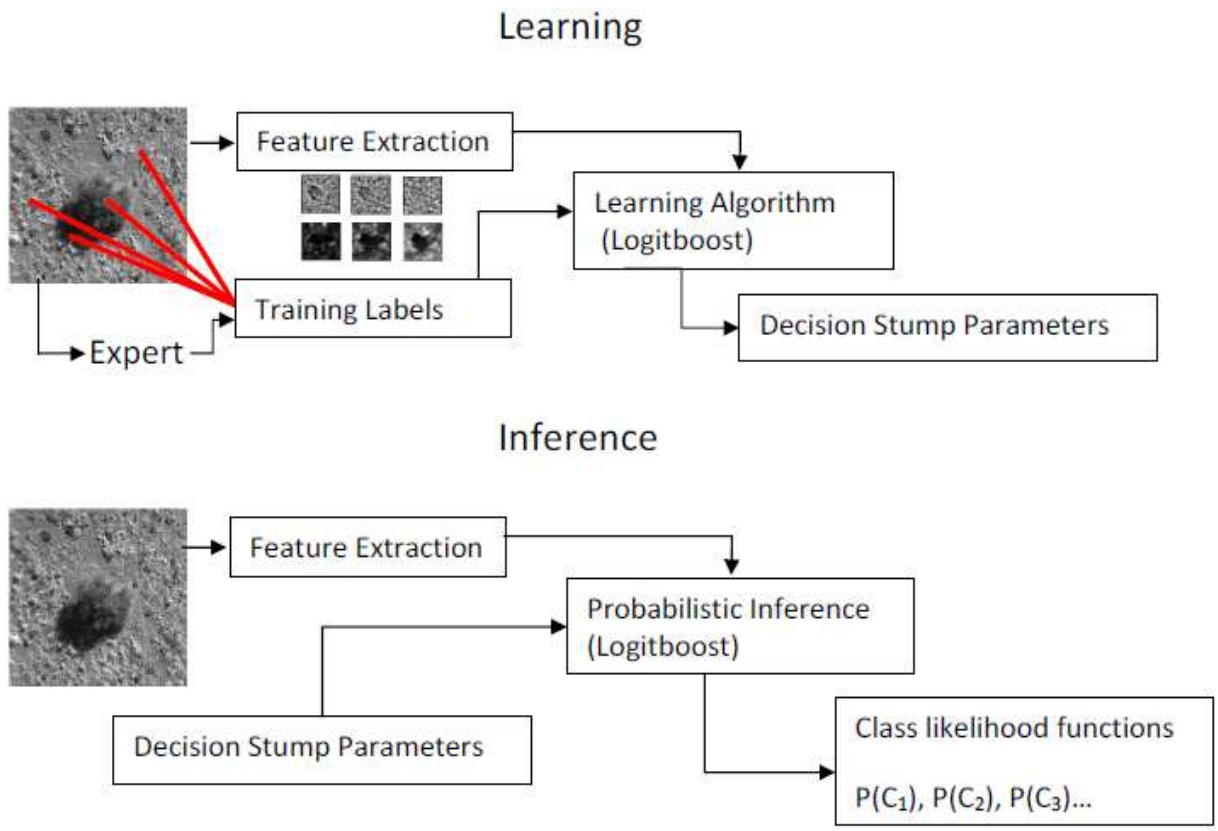

Figure 4: Block Diagram of the Implemented Classifier Training and Inference using a Machine Learning Framework with Logitboost:

class was used to group shadow, bare earth and grasses. Therefore a multi-class classification algorithm was required.

Machine learning provides an elegant approach to classification, because the patterns and ideas that a human expert uses to make a visual classification can be too complex to express formally. In a supervised machine learning framework, rather than trying to generate explicit rules or formula, the information necessary for classification can be transferred to the algorithm through a set of labelled examples. The aim of the classifier design is to best generalise this information to new unseen datasets.

Various suitable machine learning algorithms are available in the literature for classifying imagery based on texture and colour features. Neural networks are commonly used for learning and inference in texture based classification problems (Tang et al., 2003; Yu et al., 2006). Another common method for texture based classification is the Support Vector Machine (SVM), which optimises a hyperplane in order to obtain the maximum margin of classification (the maximum hyperplane distance to the nearest training datapoint). While the basic algorithm can only classify linearly separable data, SVM approaches are often extended to Kernel SVM by mapping the input dimensions to an artificially high dimensional space where linear separability can be achieved (Amari and Wu, 1999; Boser et al., 1992).

We have employed the Logitboost learning algorithm (Friedman et al., 2000) because of its natural generalisation to multiple classes, its simplicity to tune (with only two design parameters - the type of weak learner and the the number of weak learners), and its low computational complexity for learning and inference. Boosting algorithms are a family of ensemble learners that use the outputs of multiple base learning algorithms to make stronger inferences than the base algorithms could achieve individually. These weak learners do not have to be reliable or robust, but must be guaranteed to at least out-perform random guessing. Boosting theory then ensures that adding weak learners to the ensemble will better model the training data (Freund and Schapire, 1999).

Adaboost is a popular boosting algorithm proposed for binary classification (Freund and Schapire, 1999). It involves training a set of weak classifiers in rounds. A set of weights are maintained over the training examples so that difficult (most marginally classified) training cases are prioritised by further refinement. An individual learner can be optimised with respect to its error on the weighted training set, and because the cases with the lowest margin are prioritised, the learner usually generalises well in real world conditions.

While AdaBoost is relatively resilient to overfitting, it is particularly susceptible to noisy examples because it will prioritise outliers highly. In its procedural form, it does not generalise naturally to multiclass classification and instead approaches such as pair-wise classifiers have been proposed (Freund and Schapire, 1997). Instead, we employ the LogitBoost algorithm (Friedman et al., 2000) that was inspired by a statistical perspective of boosting as the process of fitting an additive model from the weak learner outputs.

In the LogitBoost algorithm, weighted outputs of weak classifiers are added to the ensemble in a forward stagewise manner to minimise a logistic loss function (Friedman et al., 2000). Each additional weak learner is optimised with respect to the loss function of the ensemble while the parameters of previous learners are held fixed. Because LogitBoost is formulated as an additive model (a weighted sum of weak learner outputs), it can be elegantly generalised to a multi-class problem by using the symmetric logistic transformation specified in (Friedman et al., 2000). In our implementation, training data is provided to the boosting algorithm as pairings $(x_1, y_1), ..., (x_n, y_n)$ where y specifies the class label, and $x_i = [x_{i1}, x_{i2}, ..., x_{id}]$ is a d dimensional feature vector extracted from each block of the imagery. We assume that if a suitable label does not exist (we are mapping a new species, or if the species has changed appearance due to seasonal factors), an expert will label a small representative fraction of the data. The weak learners we are boosting are decision stumps, one-level decision trees that threshold one of the input dimensions, a relatively common choice for boosting (Friedman et al., 2000).

When conducting inference, the LogitBoost algorithm provides likelihoods for each class, which we can use to make a classification decision. In this work, we simply took the highest likelihood class as the classification decision for every input feature set x_i.

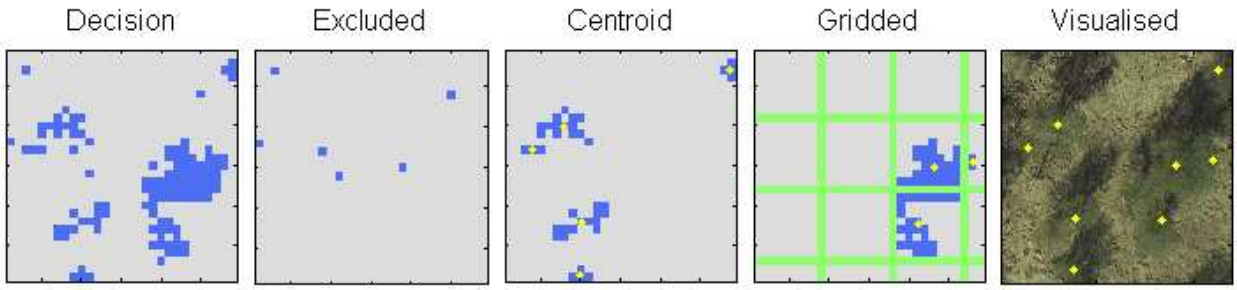

Figure 5: Example of Image Processing to Obtain a Classification Decision Bitmap: For each class, a binary one vs all decision (left) is analysed to identify connected regions and their areas. Regions of area $A < T_l$ are dismissed as noise. Regions of area $T_l \leq A < T_h$ are labelled at their centroids. Regions of area $A \geq T_h$ are partitioned by a square grid with cell area T_h, and each of the sub-regions labelled. This leads to a set of point labels over the original imagery (right).

3.3 Image post-processing

Once the classifier had been tuned and validated using the manually labelled training data, it was re-trained using all available labels (the entire training set), and applied to the rest of the imagery collected by the UAV on a per-image basis. The imagery, recorded in three channels of 1024×768 pixels, was passed through the feature extraction pipeline of Figure 3 which reduces the resolution by a factor of 16 to 64×48, and produces 27 feature dimensions (means and variances of four pyramid layers, and the values of the coarsest resolution layer, for each of the three channels).

In each frame, each of the feature vectors were independently classified to create a bitmap of class decisions (hereafter the decision bitmap). While this bitmap allows us to view the classifier output on a frame by frame basis, it is not ideal for visualisation and is not a compact representation of the data. We would rather project this data over the full terrain reconstruction, while still being able to see the colour aerial imagery and the ground truth labels. To achieve this, sparse decision labels were extracted from the decision bitmaps.

A simple heuristic procedure was implemented to identify tree crowns in the classified imagery. A binary classification (class c vs all others) was formed for each class in each image. Connected regions of these binary images were identified, and their area counted. We did not want to label very small regions such as single pixels as they were likely to be noise. On the other hand, very large regions were likely to represent many individual plants forming a connected canopy. This led to the implementation of two area thresholds T_l and T_h. Any connected region with area A such that $A < T_l$ was rejected as noise and remained unlabelled. Any region with area $T_l \leq A < T_h$ was visualised by a label at the centroid of the region. Regions with area $A \geq T_h$ were broken up by a grid of squares with area equal to T_h pixels and each of the resulting regions labelled at their centroids. This pipeline is depicted in Figure 5, and has been used to visualise classifications over the larger reconstructions in 4.3.2.

Figure 6: UAV and Sensor Payload System: Left, the J3 Cub, a small UAV used to carry the sensor payload over the designated operating area; right, the sensor payload box carried on-board the UAV consisting of a tri-axial IMU, GPS receiver, downwards-mounted colour monocular camera and PC104 computer stack for processing.

4 Results

In this section we present results from both the terrain reconstruction algorithm and vegetation classifier applied to data collected by a fixed-wing UAV operating over a large farmland area in Queensland, Australia.

4.1 Experimental Setup

This section provides an overview of the experimental setup including the flight vehicle, sensor payload used and the environment the vehicle operates within.

4.1.1 Mission Overview

Data was collected over a farmland location in Queensland Australia during several 60 minute flights of a small UAV, the J3 Cub (see Figure 6) as part of a collaborative project between the University of Sydney and Meat and Livestock Australia, with the goal of detecting and mapping infestations of weeds such as Prickly Acacia (Acacia nilotica) and Parkinsonia (Parkinsonia aculeata). These weeds cost farmers millions of dollars in damages each year due to losses of productivity and also have a significant impact on the environment by killing native vegetation and harboring feral animals. Low-cost UAV mapping in this application has the potential to provide farmers with up-to-date information on the spread of these weeds over large areas.

The UAV performed several flights, each over adjacent 4000m by 600m areas with a flight path consisting of overlapping 4km transects along the rectangular area. Logged sensor data was taken from two of the UAV flights in adjacent areas of the environment and used to

demonstrate the results of the mapping and classification algorithms presented below.

4.1.2 J3 Cub Flight Vehicle

The UAV used to carry the sensor payload is a modified one-third scale J3 Cub, capable of carrying a payload of 15kg with an endurance of one hour in its current configuration (see Figure 6). The flight vehicle has an autonomous flight control system that follows an allocated trajectory over the terrain at a fixed height of 100m above the ground.

4.1.3 Sensor Payload

Vision Camera		**IMU**	
Sampling Rate	3.75 Hz	Sampling Rate	100 Hz
FOV	$28^o \times 22^o$	Accelerometer Noise	$0.05m/s^2$ (1σ)
Resolution	1024 x 768 pixels	Gyro Noise	$0.05deg/s$ (1σ)
Angular Resolution	0.0285 deg	Accelerometer Bias Stability	$\pm 0.05m/s^2$
Pointing Direction	Downwards	Gyro Bias Stability	$\pm 0.05deg/s$

GPS Receiver	
Sampling Rate	5 Hz
Position Error	$1m$ (1σ)
Velocity Error	$10cm/s$ (1σ)

Table 1: Sensor Payload Specifications: The sensor payload consists of an IMU, GPS receiver and downwards-mounted colour monocular camera.

The vehicle carries a sensor payload consisting of a low-cost IMU, GPS receiver and a downwards-mounted colour monocular vision camera. Acceleration and rotation rate data from the IMU is sampled at 100Hz. The GPS receiver computes the earth-referenced position and velocity of the UAV at 5Hz. Colour vision frames are captured at 3.75Hz at a resolution of 1024x768 pixels. An onboard PC104 computer is used to log the sensor data, which is later processed after the vehicle lands. The specifications for each of the sensors in the sensor payload are shown in Table 1.

4.1.4 Environment and Vegetation

The operating environment is a 6x2km area of land in rural Queensland, Australia. The terrain is mostly farmland, covered by grass, shrubs and weeds. The environment contains a large population of Prickly Acacia, an invasive weeds which grows close to the river, with shrubs varying is size from about 1-3 meters tall, and also small populations of the shaggy, green weed Parkinsonia (see Figure 7) along with a variety of other Australian native trees such as Eucalyptus trees.

Figure 7: Two Examples of Invasive Weed Types found in the Mission Area: Left, a typical Prickly Acacia bush (approx. 1m tall) with fern-like leaves and large spines; right, a typical Parkinsonia bush (approx. 3m tall) with shaggy, light-green coloured stems.

4.1.5 Ground Truthing

Ground truthing data was obtained on foot using a handheld GPS (with an accuracy of approximately 5m) to record locations of particular species (see results in Section 4.3.2 for examples of collected ground truth). This data has also provided information for training human classifiers in how to interpret the imagery. Four classes were defined for this problem: Prickly Acacia (PA), Parkinsonia (PK), Eucalyptus (EUC) and a fourth class (NULL) to represent any grass, mud, shadows or other background features that are not of interest. A GUI was developed for a user to manually identify 16x16 blocks containing examples of the above classes. Approximately 1000 blocks were identified and labelled over all of the visual data, to form an example library for training and cross validation. Frames of imagery were selected uniformly, and interesting features selected from the images by a human. The number of examples of each class reflect the relative abundance of each class in the data (see Table 2).

Class	Number of Examples
PA	248
PK	67
EUC	181
NULL	544

Table 2: Number of Training Examples used in Classification for the 4 Different Classes.

4.2 Trajectory and Map Reconstruction Results

This section presents results from the UAV trajectory and map reconstruction system.

4.2.1 Vision Feature Extraction and Matching Results

Figure 8 illustrates an example of the feature tracking process described in Section 2.2 over two frames. Figure 8 (a) shows the original, undistorted image data for the first frame and Figure 8 (b) shows the original, undistorted image data for the second frame. The overlap between the frames is high (each image shares about 80% of the field of view in common). Figure 8 (c) illustrates the corner points extracted from the first image. Figure 8 (d) shows the tracked positions of the corner points in the second frame using the LK tracker, where these correspondences are used to compute the fundamental matrix using a RANSAC method. Figure 8 (e) illustrates the feature tracks that are validated by the epipolar constraints derived from the fundamental matrix, where Figure 8 (f) illustrates the feature tracks that are rejected as outliers. The tracks are illustrated by a line drawn between the original point feature location and the tracked point feature location, but shown on the one image (i.e. the optical flow of the feature tracks).

It can be seen from the illustration of accepted feature matches in Figure 8 (e), that the optical flow of the features has been estimated consistently across the image; the tracks account for the forward motion along with a small amount of rolling rotation of the UAV; this type of optical flow is consistent with other frames throughout the flight.

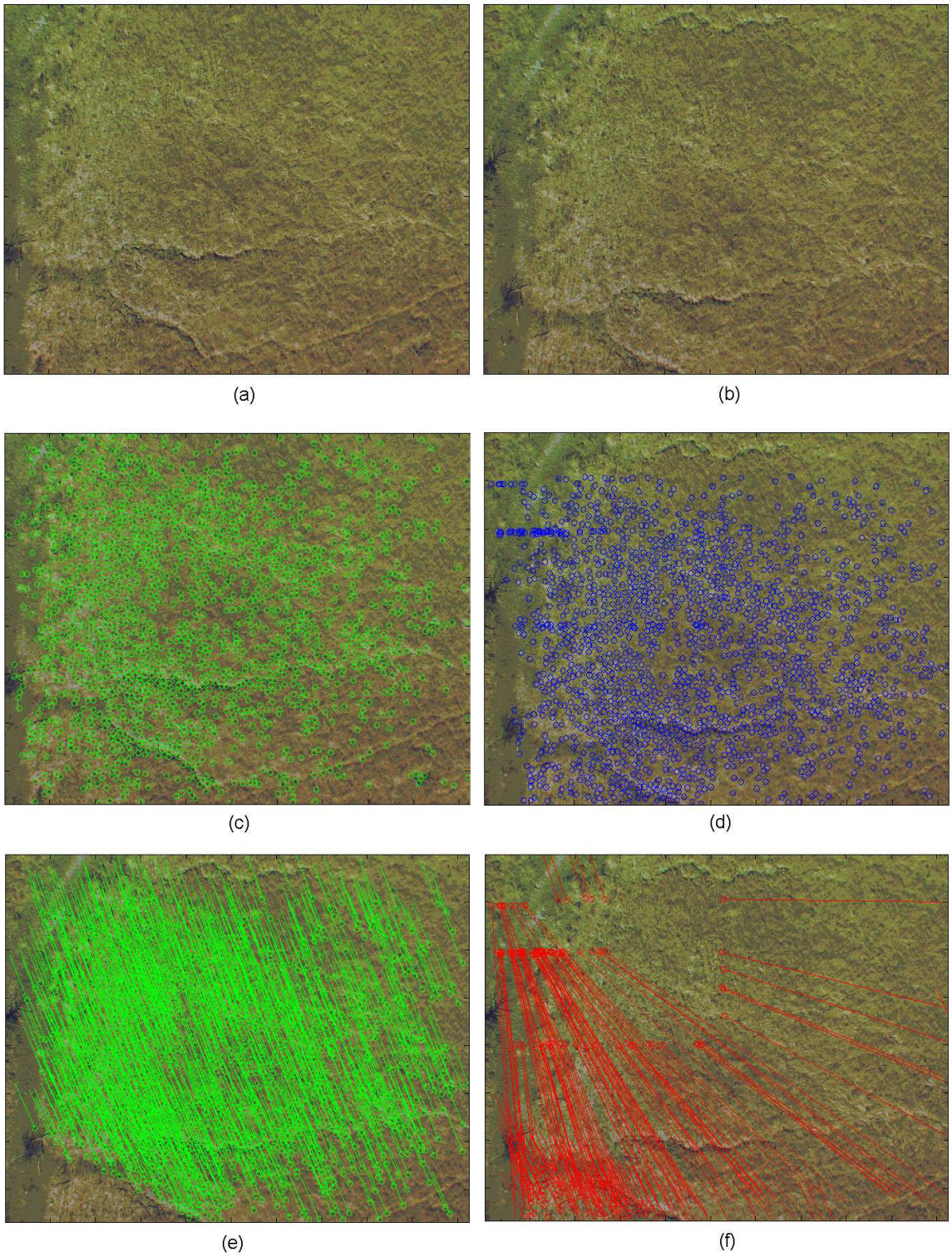

Figure 8: Example of Vision Feature Extraction and Matching Process: (a) the original image for frame 1, (b) the original image for frame 2, (c) features extracted from frame 1, (d) extracted feature locations tracked from frame 1 to frame 2 using an LK tracker, (e) correct feature matches illustrated as optical flow lines within frame 1, (f) outlier matches from the LK tracker, detected and rejected using epipolar constraints.

4.2.2 UAV Trajectory and Map Bundle Adjustment Results

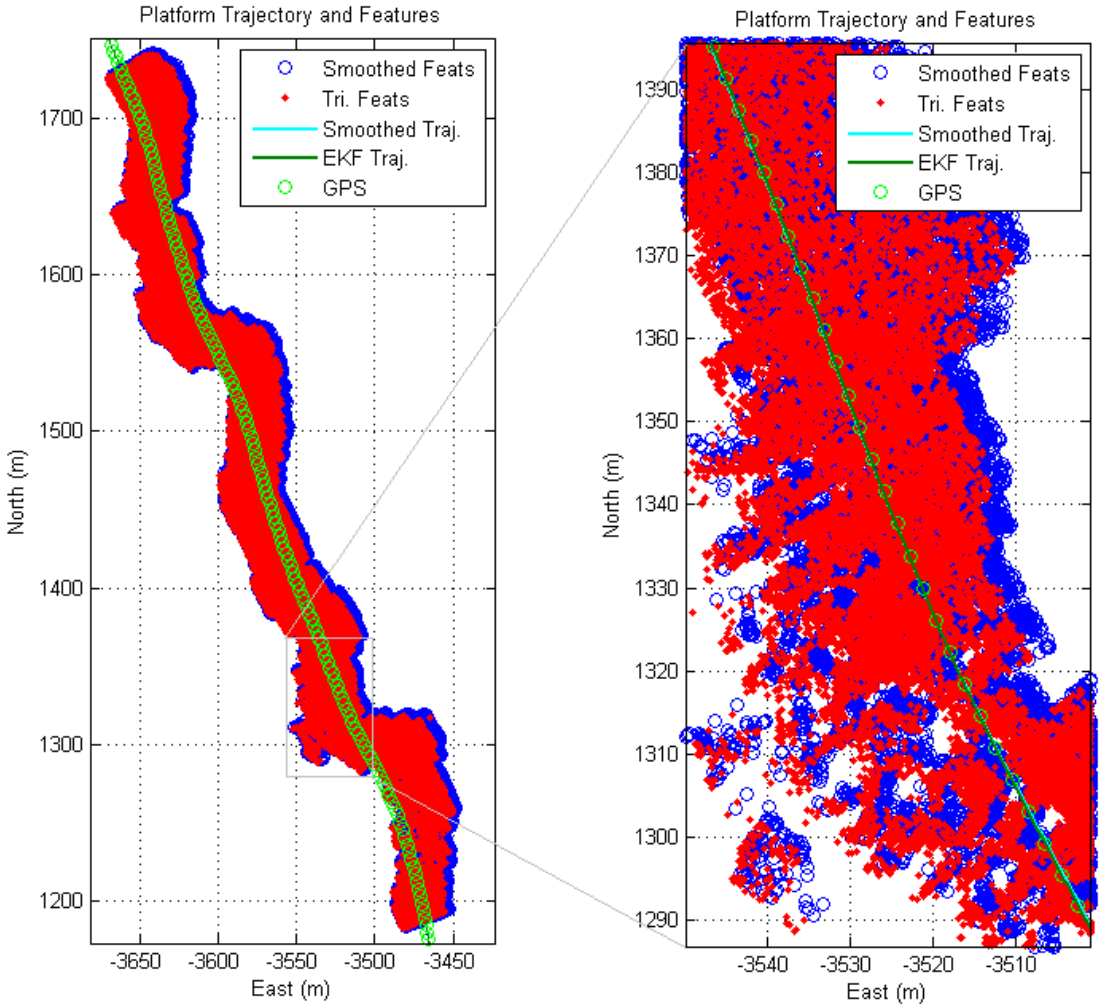

Figure 9: Comparison of Initial Map Feature Estimates and Bundle Adjusted Map Feature Estimates: Shown in red are the initial map feature estimates calculated through triangulation using EKF derived pose information (see Section 2.4.2). Shown in blue are the final map feature estimates after the bundle adjustment procedure (see Section 2.5). Also shown are the initial and final estimates of the UAV trajectory over the map and GPS position observations.

Figure 9 shows an overhead view of both the initial estimates and final bundle adjusted UAV trajectory and map feature points for an isolated section of the first flight trajectory and terrain map. Also shown are the GPS position observations. In this example, the bundle adjustment process was applied over an isolated section of the UAV trajectory and the terrain encompassing about 3350 UAV poses and 105000 terrain point features. The bundle adjustment procedure takes 16 iterations to converge from the initial estimate, where each iteration for this section of the data takes approximately 8-9 seconds using a Intel Core2Duo T7250 2GHz processor with 2Gb RAM.

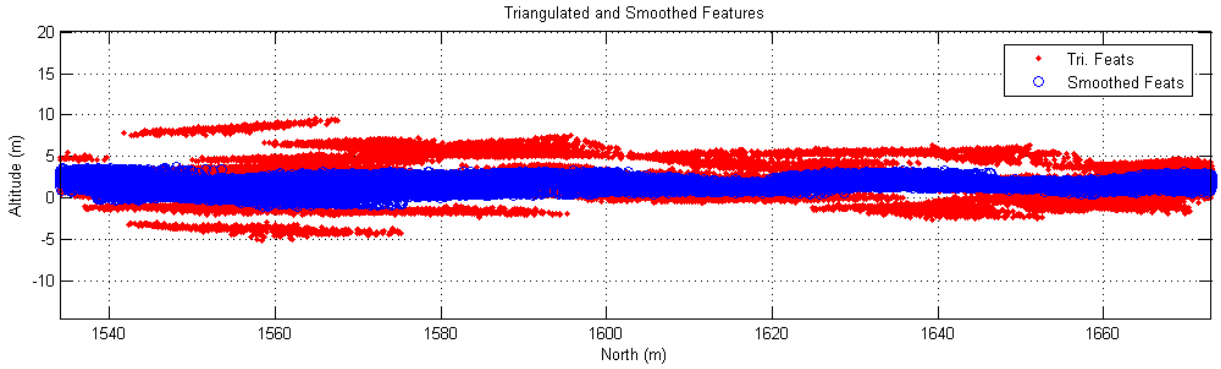

Figure 10: Comparison of Vertical Map Feature Estimates in the Initial and Final Bundle Adjusted Maps: Shown is a north vs. altitude plot of the initial map feature estimates (shown in red) and the final map feature estimates after the bundle adjustment procedure (shown in blue). Prior to the bundle adjustment procedure, there is a large deviation in the height of features between different frames based mainly on errors in the alignment angles of the camera, UAV attitude errors and triangulation errors. After bundle adjustment, the errors in the terrain elevation are reduced and a more self-consistent terrain map produced, from which the height of the terrain can be reliably extracted for use in mosaicing.

Figure 10 shoes a vertical perspective (north vs. altitude plot) of both the initial estimates and final bundle adjusted map feature points for the same isolated section of the flight trajectory and terrain map. Prior to the bundle adjustment procedure, there is a large deviation in the height of features between different frames based mainly on errors in the alignment angles of the camera, UAV attitude errors and triangulation errors. After bundle adjustment, the errors in the terrain elevation are reduced and a more self-consistent terrain map produced, from which the height of the terrain can be reliably extracted for use in mosaicing.

4.2.3 Map Mosaicing Results

Figure 11 shows results of the mosaicing algorithms presented in Section 2.6. Shown in the left hand side of the figure is the entire 4000 by 600 meter area mapped out by the UAV during the first flight. The mosaic is built from approximately 14000 images. The flight path follows along a dry river bed where an abundance of different vegetation is found including both native trees and invasive species. The imagery collected covers most of the area where the small gaps seen in the map occur due to gaps in the coverage of the sensor footprint that occur when the vehicle roll side-to-side due to occasional wind disturbances. For the most part, the mosaic is constructed from the initial UAV trajectory and map estimates as described in Section 2.6 where as the isolated section shown in red has been built using the refined, bundle adjusted estimates. Shown in the right hand side of the figure is a zoomed-in view of the section of the mosaic where bundle adjusted estimates are used (corresponding to the same area seen in Figure 9 above). The zoomed-in view clearly illustrates features of the terrain such as parts of the dry river bed, grass and different trees.

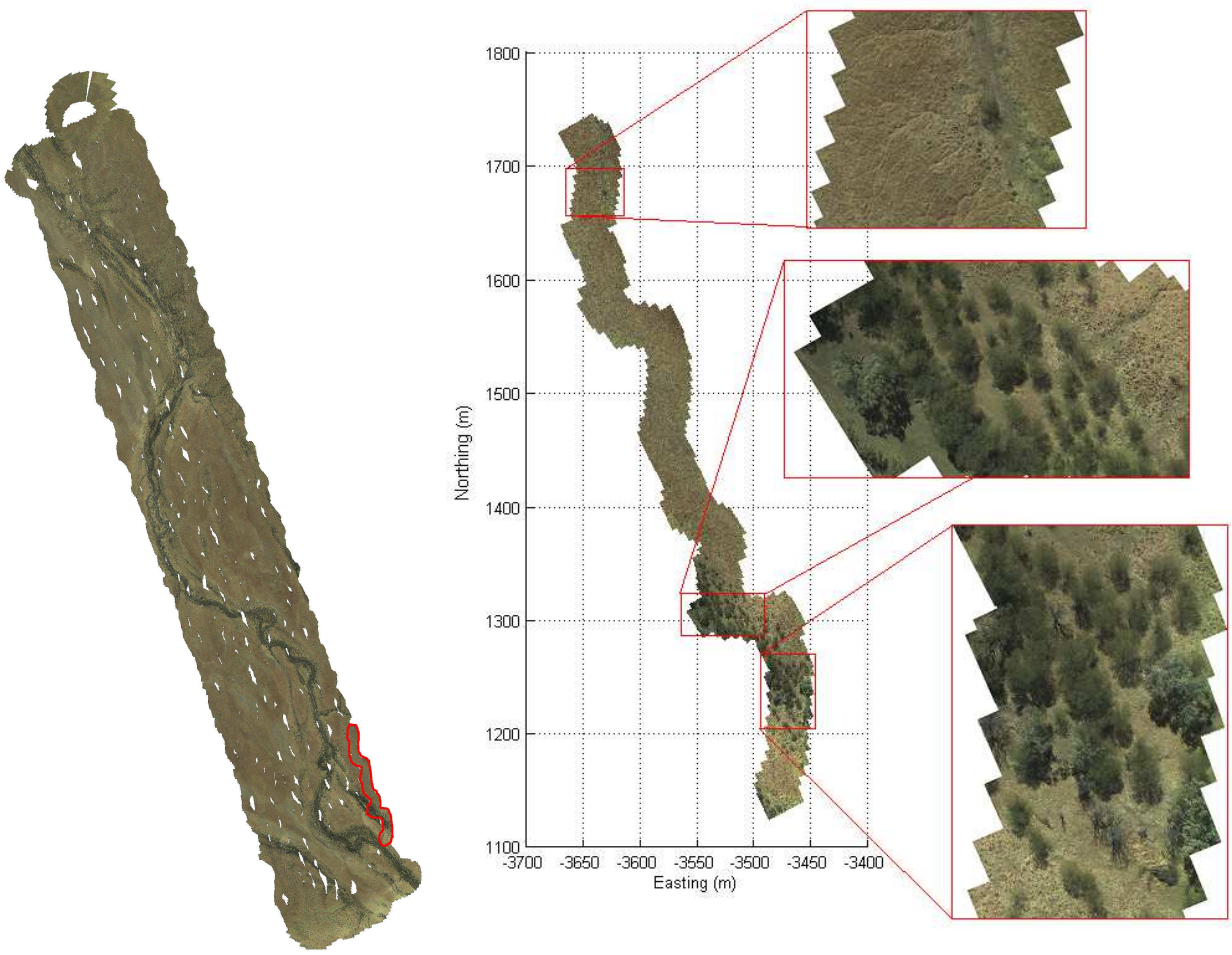

Figure 11: Final Environment Mosaic: Left, mosaic map of the entire first flight area covering a distance of approximately 4000 by 600 meters. Right, zoomed in section of the mosaic map showing areas covered by grass, trees and different types of vegetation. The terrain mapping and mosaicing systems produce a consistent, geo-referenced map of the environment which can be used for environmental monitoring.

Figure 12 shows a section of the constructed mosaic and a comparison between the mosaic constructed using initial UAV trajectory and map estimates (before bundle adjustment) and the final UAV trajectory and map estimates (after bundle adjustment). The unadjusted mosaic, although generally spatially correct, shows signs of poorly aligned imagery and inconsistent mapping due mainly to the errors in the camera to IMU alignment angles, UAV attitude estimates and poor terrain altitude estimation. The final bundle adjustment mosaic, demonstrates well aligned imagery and a consistent map (even without any type of texture blending) due to the improved accuracy of UAV trajectory and map point estimates in the bundle adjustment process.

Figure 12: Comparison of Mosaics for Initial and Final UAV Trajectory and Terrain Map Estimates: (a) Section of environment mosaic generated using the initial UAV trajectory and map feature point estimates before bundle adjustment and, (b) same section of environment mosaic generated using the final estimates after bundle adjustment. The unadjusted mosaic shows signs of poorly aligned imagery and inconsistent mapping due to mainly to the errors in the camera to IMU alignment angles, UAV attitude estimates and poor terrain altitude estimation. The final bundle adjustment mosaic, demonstrates well aligned imagery and a consistent map (even without any type of texture blending) due to the improved accuracy of UAV trajectory and map point estimates in the bundle adjustment process.

4.3 Vegetation Classification Results

This section presents results from the classification algorithms for distinguishing between different types of vegetation based on their appearance in the aerial imagery.

4.3.1 Cross Validation

A 20 fold cross validation, with randomly assigned folds of 52 examples, was run on the training data. Our Logitboost/Decision Stump classifier has only one tuning parameter - the number of weak learners. It is expected that too few learners will give poor generalisation and inference as there is not enough flexibility in the model to capture sufficient information from the training inputs. On the other hand, too many learners may lead to model over-fitting and be detrimental to inference in unseen data. To evaluate this relationship, the classifier error rate (correct vs incorrect class selection) was plotted against number of learners (Figure 13). The error rate essentially peaks at 150 stumps, and does not improve as more are added (despite an improvement in the training set performance).

In this data, vision based classification is fundamentally difficult, and some of the imagery is hard even for a human to classify. Therefore the resulting 20% error rate is relatively good, especially for a 4 class problem. The confusion matrix (Figure 14) details the nature of the

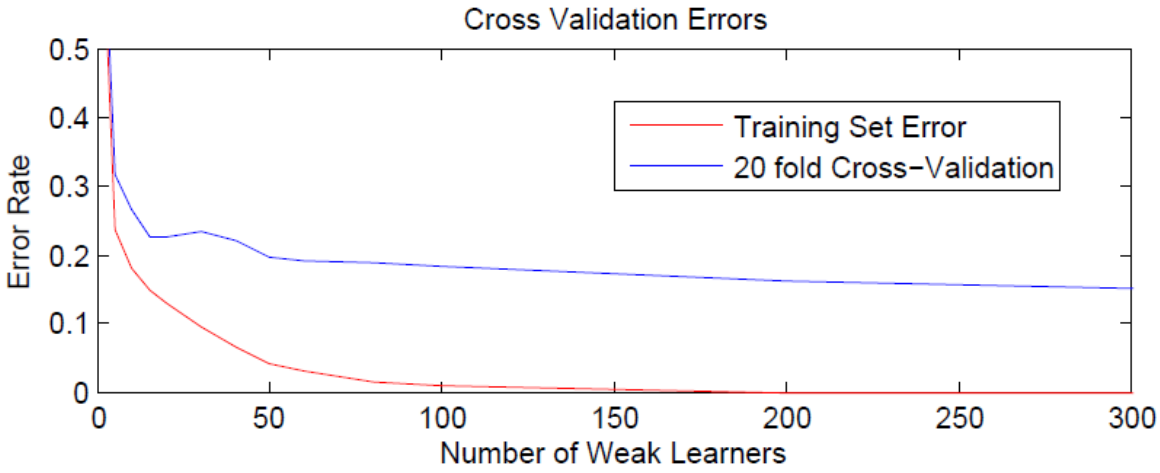

Figure 13: Performance of the classifier in cross validation vs the number of weak learners (decision stumps) used. Beyond 100 stumps, the classifier improves performance on the training set without any improvements in generalisation to the testing set. The performance limit of the current implementation appears to be a 20% error rate.

classification error when using 150 stumps. The precision (fraction of positive returns that are correct) for each class and the recall (fraction of each class that is detected) are also provided along with an F_1 score, which is the harmonic mean of the precision and recall and provides a measure of classification accuracy where 1 is the most accurate and 0 is the least.

The confusion matrix in Figure 14 shows that the classifier is effective at identifying vegetation/non vegetation (as shown by the high precision and recall of the null class). Likewise, it is effective at distinguishing between the native Eucalypts and the two woody weed species in the data, because of the distinctive appearance of the Eucalyptus.

The classifier is less reliable at distinguishing between the two woody weeds Prickly Acacia (PA) and Parkinsonia (PK) due to their similar appearance in the aerial imagery. Because of the relative abundance of PA in the training and testing data (and under the flight path), greater overall reliability is achieved by classifying the marginal cases as PA. Consequently the classifier has mis-labeled only 18 of the 248 PA examples as PK, while 24 of the 67 PK were labeled PA. If we wanted to improve the recall of PK at the expense of overall classifier precision, we could apply a different decision to the logitboost outputs (instead of taking the highest class probability, we could make a decision that favours PK in the marginal cases for example).

4.3.2 Classified Map

A section of imagery collected by the UAV was classified, and the extracted labels were plotted over a mosaiced colour image. Figure 15 illustrates both a section of the raw constructed map during the second flight of the UAV and an overlay of classified regions in the image

Predicted \ Actual	PA	PK	EUC	NULL
PA	208	24	2	23
PK	18	26	1	10
EUC	1	0	152	28
NULL	20	17	26	483

	PA	PK	EUC	NULL
Precision	0.81	0.47	0.84	0.88
Recall	0.84	0.39	0.84	0.89
F-Score	0.82	0.43	0.84	0.88

Figure 14: Left: Confusion matrix for the classifier output in 20 fold cross validation using 150 stumps. Right: Corresponding precision and recall statistics. High precision and recall values for the PA, EUC and NULL classes indicate that the classifier can reliably identify Parkinsonia and Eucalypt, and is also effective at seperating tree from non-tree. However, the classifier often mis-labels the PK class as the more abundant PA class due to its similarity in visual appearance (differentiation is even difficult for a human) leading to poor scores for the PK class.

corresponding to different species of vegetation. Figure 16 illustrates a zoomed-in section of the classified map and a comparison to ground survey data of the vegetation collected in the area. Shown with the collected ground truth data is the associated 5m error ellipse corresponding to the errors in the handheld GPS receiver used to collect the ground truth data.

In the mosaiced classification data, tree crowns are easy for an observer to identify, and the classifier has also reliably labelled them with one of the three tree classes. As was expected from the confusion matrix results of Figure 14, the classifier is effective at identifying Eucalypts in the data, probably because its appearance is fairly distinctive in the imagery. We also know (from ground truthing) that the vast majority of vegetation in the dataset should be labelled as Prickly Acacia (PA). While most of the Acacia is correctly labelled, the classifier clearly has difficulty distinguishing between the two woody weeds Prickly Acacia (PA) and Parkinsonia (PK) in the resolution of data available. Incorrectly classified plants are often partly labelled correctly and partly incorrectly, suggesting there is scope for improvement from using an object based approach to aggregate features over tree crowns (making species identification a separate decision from segmentation). Despite the difficulties faced, the classification algorithm has managed to separate the collected information into the correct classes in most cases, where the number of incorrect classifications is small enough to allow a human weed expert to correct for outliers.

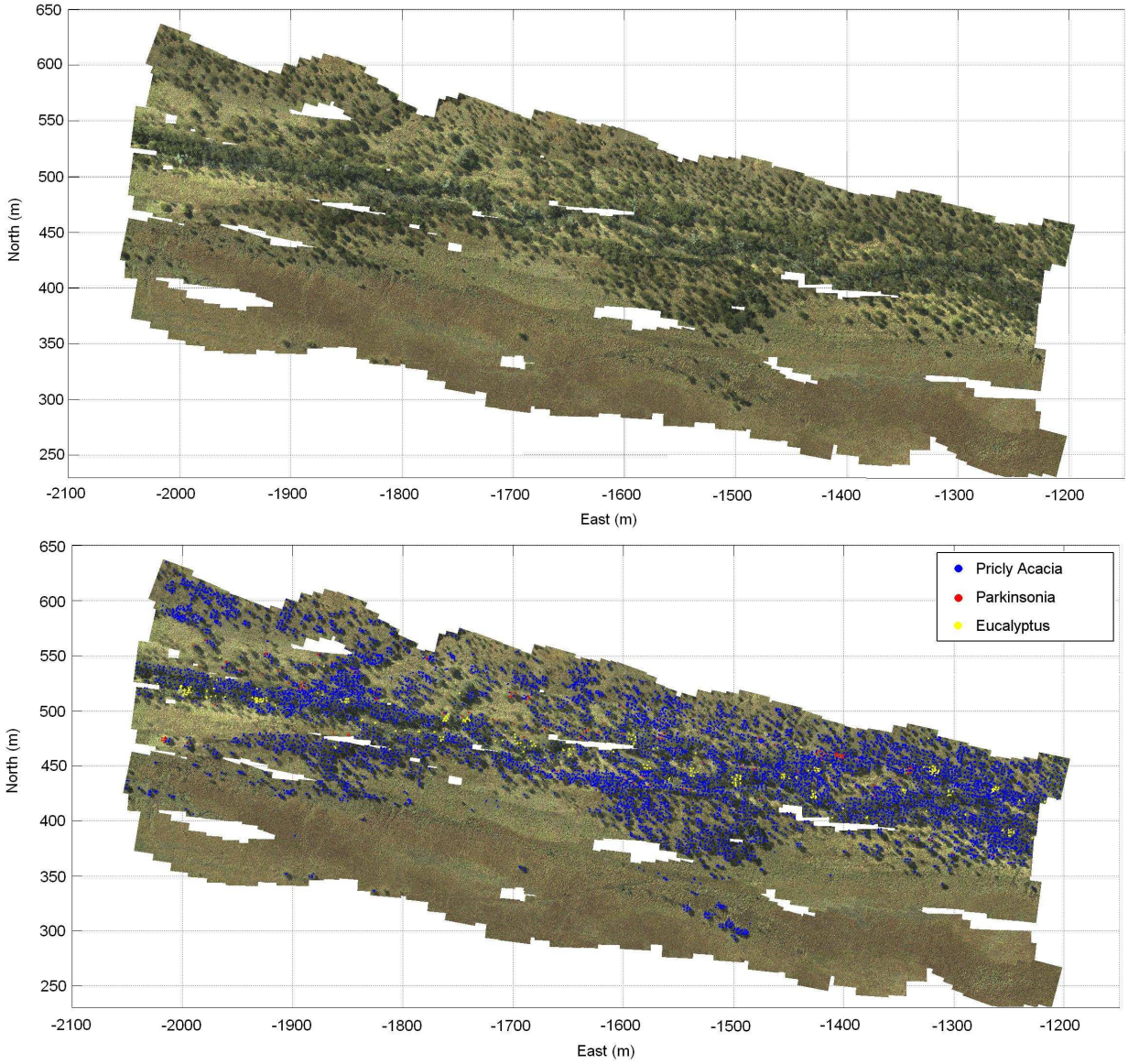

Figure 15: Mosaic Map of Classified Vegetation: Shown in the upper image is the raw mosaic map constructed using the collected vision data and UAV trajectory and map feature estimates over a selected section of the second flight area. Shown in the lower image is the same mosaic with classified vegetation overlayed, corresponding to two different weeds (Prickly Acacia and Parkinsonia) and one native tree (Eucalyptus).

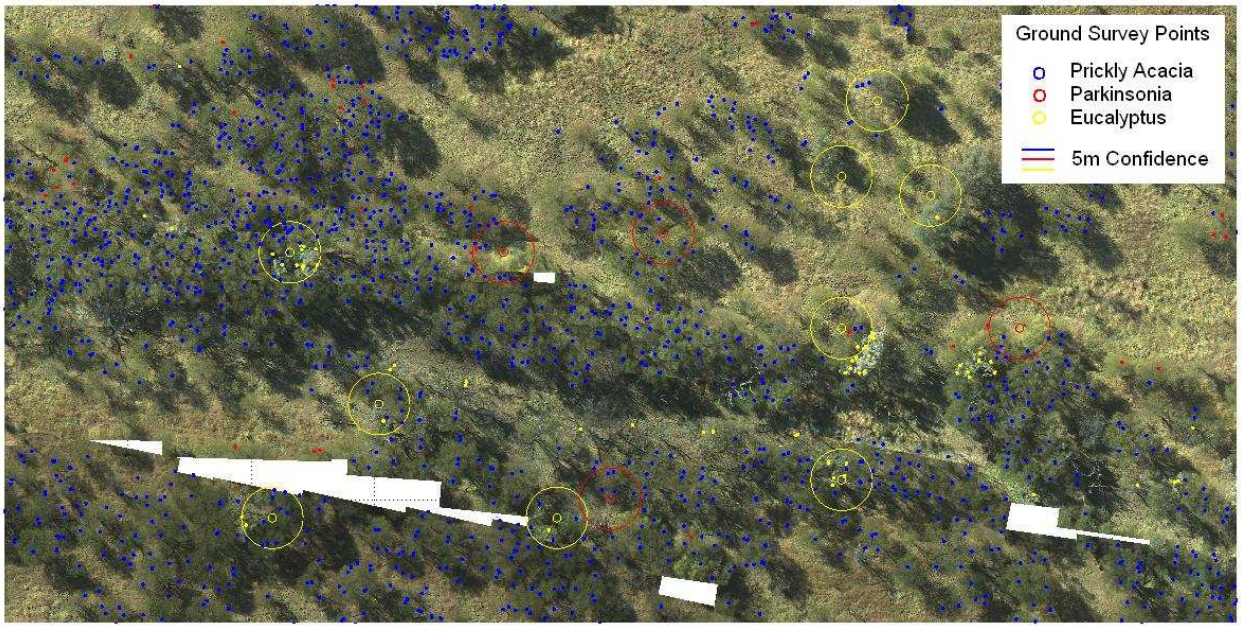

Figure 16: Mosaic Map of Classified Vegetation (Zoomed-in with Ground Survey Overlay): Shown is a zoomed-in section of the classified map showing examples of the three main classes of vegetation. The large, sparse circular points indicate the positions and 5m confidence interval of vegetation identified during a ground survey of the area.

5 Conclusions and Future Work

This paper has presented a framework for integrating low-cost sensor information from an IMU, GPS receiver and monocular vision camera for building large-scale terrain reconstructions and classifying different types of vegetation in the environment from an aerial vehicle. The fusion of IMU and GPS information with the vision data allows for the estimation of a geo-referenced terrain map that is fixed in translation, rotation and scale. Using generic monocular vision-based colour and texture descriptors, different types of vegetation visible in the image data can be distinguished and classified owing to the low-altitude of the UAV flight. A classification algorithm is implemented based on supervised learning of examples of different types of vegetation that are provided by a human expert; the resulting algorithm learns the connection between different classes of vegetation and the vision descriptors. The classified vegetation is then geo-referenced using the final terrain map and presented to the user.

The final classification results indicate that the classifier has performed well at distinguishing woody weeds from native vegetation, however has difficultly in fine discrimination between different species of weed, due to their similar appearance and relatively small number of training examples available. These different species are known to have differing tree shape and size properties which could be used in future work as extra features for classification. This would require even a higher level of density in 3D terrain reconstruction, perhaps at the pixel-level of the camera, to identify the shape and roughness of different vegetation. One avenue for future work will examine the applicability of dense stereo-vision methods for providing this detail over areas where the classification has located weed species. Future work will also focus of integrating other spatial information from the map (such as the relative position of different vegetation w.r.t one another and other landscape features such as river-beds etc.) to help improve the distinguishable characteristics of difference species.

It is difficult to speculate on the flexibility of our current classifier to changes in lighting conditions or the seasonal appearance of the vegetation, as time series data is not easily obtained. If conditions are significantly different, then the same algorithm may be used, but new training examples must be provided. Future work will examine the effectiveness of the proposed scheme for data gathered in a variety of different lighting conditions.

Isolated sections of the results from the experimental data show areas of missed coverage and areas where classification results could be improved by additional quantities of image data. The application thus demonstrates the benefit for adaptive methods in data collection, which will be explored in future work. Adaptive data collection requires methods for online terrain reconstruction and classification where the map is incrementally built while the UAV is in flight, rather than being performed in a post-processing step. This would then allow the UAV to target it's search towards areas of missed coverage and areas of weed infestation, where the distinction between different weed species is critical.

Acknowledgments

This work is supported in part by Meat and Livestock Australia (MLA) under project code B.NBP.0474, ''UAV Surveillance Systems for the Management of Woody Weeds". This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

Alchanatis, V., Ridel, L., Hetzroni, A., and Yaroslavsky, L. (2005). Weed detection in multi-spectral images of cotton fields. *Computers and Electronics in Agriculture*, 47(3):243–260.

Amari, S. and Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789.

Amestoy, P., Davis, T., and Duff, I. (1996). An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4):886–905.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J. D., Hammarling, A. G. S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition.

Bajwa, S. and Tian, L. (2001). Aerial CIR remote sensing for weed density mapping in a soybean field. *Transactions of the American Society of Agricultural Engineers*, 44(6):1965–1974.

Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *5th Annual ACM Workshop on COLT*, pages 144–152. ACM Press.

Bouguet, J. (1st October, 2009). Camera Calibration Toolbox for MATLAB. In *Retreived from http://www.vision.caltech.edu/bouguetj/calib_doc/*.

Bryson, M., Johnson-Roberson, M., and Sukkarieh, S. (2009). Airborne Smoothing and Mapping using Vision and Inertial Sensors. In *IEEE International Conference on Robotics and Automation*.

Bryson, M. and Sukkarieh, S. (2007). Building a Robust Implementation of Bearing-Only Inertial SLAM for a UAV. *Journal of Field Robotics, Special issue on SLAM in the field*, 24(2):113–143.

Burt, P. and Adelson, T. (1983). The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 9(4):532–540.

Carleer, A. and Wolff, E. (2004). Exploitation of very high resolution satellite data for tree species identification. *Photogrammetric Engineering & Remote Sensing*, 70(1):135–140.

Casady, G., Hanley, R., and Seelan, S. (2005). Detection of leafy spurge (euphorbia esula) using multidate high-resolution satellite imagery. *Weed technology*, 19 (2):462–467.

Chang, T. and Kuo, C. J. (1993). Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, 2:4.

Clark, R., Lin, M., and Taylor, C. (2006). 3D Environment Capture from Monocular Video and Inertial Data. In *SPIE on Three-Dimensional Image Capture and Applications*.

Culvenor, D. (2002). TIDA: an algorithm for the delineation of tree crowns in high spatial resolution remotely sensed imagery. *Computers & Geosciences*, 28:33–44.

Czaplewski, R. and Patterson, P. (2004). Classification accuracy for stratification with remotely sensed data. *Forest Science*, 49(3):402–408.

Ehlers, M., Gahler, M., and Janowsky, R. (2003). Automated analysis of ultra high-resolution remote sensing data for biotope type mapping: New possibilities and challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*, 57(5-6):315–326.

Erikson, M. and Olofsson, K. (2005). Comparison of three individual tree crown detection methods. *Machine Vision and Applications*, 16(4):256–265.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer System Sciences*, 55(1):119–139.

Freund, Y. and Schapire, R. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407.

Glenn, N., Mundt, J., Weber, K., Prather, S., Lass, W., and Pettingill, J. (2005). Hyperspectral data processing for repeat detection of small infestations of leafy spurge. *Remote Sensing of the Environment*, 95:399–412.

Guerschman, J., Paruelo, J., Bella, C., Giallorenzi, M., and Pacin, F. (2003). Land cover classification in the Argentine Pampas using multi-temporal Landsat TM data. *Remote Sensing*, 24(17):3381–3402.

Harvey, K. and Hill, G. (2001). Vegetation mapping of a tropical freshwater swamp in the Northern Territory, Australia: A comparison of aerial photography, Landsat TM and SPOT satellite imagery. *International Journal of Remote Sensing*, 22(15):2911–2925.

Heeger, D. and Bergen, J. (1995). Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd International Conference on Computer Graphics and Interactive Techniques*.

Hsieh, P., Lee, L., and Chen, N. (2001). Effect of spatial resolution on classification errors of pure and mixed pixels in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 39(12):2657–2663.

Johnson-Roberson, M., Pizarro, O., Williams, S., and Mahon, I. (2009). Generation and Visualization of Large Scale 3D Reconstructions from Underwater Robotic Surveys. *To Appear in Journal of Field Robotics*.

Kim, J., Wishart, S., and Sukkarieh, S. (2003). Real-Time Navigation, Guidance and Control of a UAV using Low-cost Sensors. In *4th International Conference on Field and Service Robotics*.

Klinken, R., Shepherd, D., Parr, R., Robinson, T., and Anderson, L. (2007). Mapping mesquite (prosopis) distribution and density using visual aerial surveys. *Rangeland Ecology Management*, 60:408–416.

Koch, R., Pollefeys, M., and Gool, L. V. (1998). Multi Viewpoint Stereo from Uncalibrated Video Sequences. In *5th European Conference on Computer Vision*.

Lamb, W. and Lamb, R. (2002). Evaluating the accuracy of mapping weeds in seedling crops using airbourne digital imaging. *Weed Research*, 39(6):481–492.

Lawes, R. and Wallace, J. (2008). Monitoring an invasive perennial at the landscape scale with remote sensing. *Ecological Management and Restoration*, 9(1):53–58.

Li, S. and Shawe-Taylor, J. (2005). Comparison and fusion of multiresolution features for texture classification. *Pattern Recognition Letters*, 26:633–638.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Imaging Understanding Workshop*.

Madden, M. (2009). GeoEye-1, the world's highest resolution commercial satellite, optical society of america. In *Conference on Lasers and Electro-Optics*, number PWB4.

Manjunath, B. and Madden, W. (1996). Texture features for browsing and retrieval of large image data. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 18(8):837–842.

McCarthy T., F. A. and G, O. (2007). Compact Airborne Image Mapping System (CAIMS). In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Medlin, C., Shaw, D., Gerard, P., and LaMastus, F. (2000). Using remote sensing to detect weed infestations in Glycine max. *Weed Science*, 48(3):393398.

Mostafa, M. and Schwarz, K. (2000). A Multi-Sensor System for Airborne Image Capture and Georeferencing. *Photogrammetric Engineering and Remote Sensing*, 66(12):1417–1423.

Nagendra, H. and Rocchini, D. (2008). High resolution satellite imagery for tropical biodiversity studies: the devil is in the detail. *Biodiversity and Conservation*, 17(4):3431–3442.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer.

Pinies, P., Lupton, T., Sukkarieh, S., and Tardos, J. (2007). Inertial Aiding of Inverse Depth SLAM using a Monocular Camera. In *Internation Conference on Robotics and Automation*.

Pollefeys, M. (2004). Visual Modelling with a Hand-Held Camera. *International Journal of Computer Vision*, 59(3):207–232.

Qian, G., Zheng, Q., and Chellappa, R. (2000). Reduction of inherent ambiguities in structure from motion problem using inertial data. In *IEEE International Conference on Image Processing*.

Robinson, T. and Metternicht, G. (2005). Multi-temporal spatial modelling of noxious weed distribution using historic remote sensing imagery. In *Proceedings of International Cartographic Conference*.

Samal, A., Brandle, J., and Zhang, D. (2006). Texture as the basis for individual tree identification. *Information Sciences*, 176(5):565–576.

Sandmann, H. and Lertzman, K. (2003). Combining high-resolution aerial photography with gradient-directed transects to guide field sampling and forest mapping in mountainous terrain. *Forest Science*, 49(3):429–443.

Shi, J. and Tomasi, C. (1994). Good Features to Track. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Simoncelli, E. and Freeman, W. (1995). The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *2nd IEEE International Conference on Image Processing*, volume 3, pages 444–447.

Sukkarieh, S., Nettleton, E., Kim, J., Ridley, M., Goktogan, A., and Durrant-Whyte, H. (2003). The ANSER Project: Data Fusion Across Multiple Uninhabited Air Vehicles. *International Journal of Robotics Research*, 22(7):505–539.

Sun, W., Heidt, V., Gong, P., and Xu, G. (2003). Information fusion for rural land-use classification with high resolution satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4):883–890.

Tang, L., Tian, L., and Steward, B. (2003). Classification of broadleaf and grass weeds using Gabor wavelets and an artificial neural network. *Trans. ASAE*, 46(4):1247–1254.

Tang, L., Tian, L., Steward, B., and Reid, J. (1999). Texture-based weed classification using gabor wavelets and neural network for real-time selective herbicide applications. Technical report, American Society of Agricultural Engineers, Michigan.

Titterton, D. and Weston, J. (1997). *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd., London.

Torr, P. and Murray, D. (1997). The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. *International Journal of Computer Vision*, 24(3):271300.

Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (2000). *Bundle Adjustment - A Modern Synthesis: In Vision Algorithms: Theory and Practice*. Springer Verlag.

Ye, X., Sakai, K., Asada, S.-I., and Sasao, A. (2007). Use of airbourne multispectral imagery to discriminate and map weed infestations in a citrus orchard. *Weed Biology and Management*, 7(1):23–30.

Yu, Q., Gong, P., Clinton, N., Biging, G., Kelly, M., and Schirokauer, D. (2006). Object-based detailed vegetation classification with airbourne high spatial resolution remote sensing imagery. *Photogrammetric Engineering and Remote Sensing*, 72(7):799–811.

Zhang, N., Wang, M., and Wang, N. (2002). Precision agriculture - a worldwide overview. *Computers and Electronics in Agriculture*, 36:113–132.