# Unaided Stereo Vision Based Pose Estimation

**Michael Warren**
University of Queensland, Australia
m.warren1@uq.edu.au

**David McKinnon**
Queensland University of Technology, Australia
david.mckinnon@qut.edu.au

**Hu He**
University of Queensland, Australia
h.hu2@uq.edu.au

**Ben Upcroft**
University of Queensland, Australia
ben.upcroft@uq.edu.au

## Abstract

This paper presents the development of a low-cost sensor platform for use in ground-based visual pose estimation and scene mapping tasks. We seek to develop a technical solution using low-cost vision hardware that allows us to accurately estimate robot position for SLAM tasks. We present results from the application of a vision based pose estimation technique to simultaneously determine camera poses and scene structure. The results are generated from a dataset gathered traversing a local road at the St Lucia Campus of the University of Queensland. We show the accuracy of the pose estimation over a 1.6km trajectory in relation to GPS ground truth.

## 1 INTRODUCTION

This paper presents the development of a robotic sensor platform to gather data for the application of unaided visual pose estimation. The platform is currently used for logging tasks to perform subsequent offline processing, but has the eventual aim of online pose estimation and mapping. Our intention is to use the platform for handheld pose estimation in both indoor and outdoor tasks and pose estimation from both ground and airborne robotic platforms. We also intend to use the results for future dense scene reconstruction tasks.

Accurate pose estimation for robotic platforms is essential for precise mapping and localisation tasks requiring for example geo-located/metric positioning information or detailed 3D scene reconstruction. Unaided visual pose estimation has seen rapid improvements enabling localisation in large-scale environments in the absence of sensors such as GPS. The key technology for constructing a map and maintaining precise pose estimates is to use visual motion registration between image frames [Konolige and Agrawal, 2008]. The best estimate can then be computed using bundle adjustment [Triggs *et al.*, 2000], a nonlinear optimisation over the pose states of the vehicle and visually observed features.

In this paper, we describe a sensor platform able to acquire high rate visual data which is subsequently processed using bundle adjustment. We illustrate the power of these visual localisation and mapping methods over a large-scale and complex environment.

We primarily use off-the-shelf hardware and open source software in developing the sensor platform resulting in an inexpensive system. Such hardware has already been well tested, of high quality and usually of small form factor. Such properties are essential for a reliable robotic platform and its use in small size, low weight requirement platforms such as ground based indoor robots and Unmanned Aerial Vehicles.

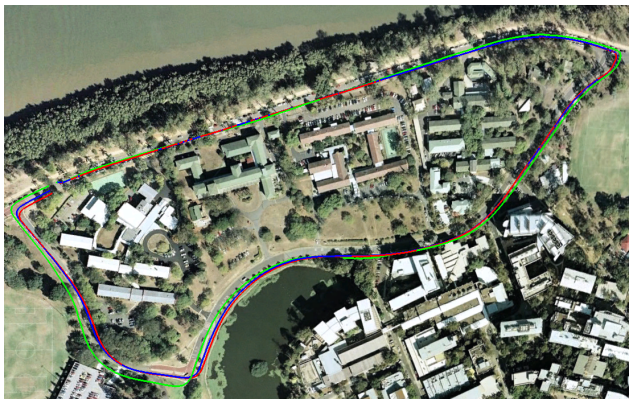

Figure 1: Comparison of unaided visual pose estimates (non-bundle adjusted trajectory in red, bundle adjusted trajectory in blue) with GPS (green) on data gathered by the platform overlaid on a satellite image.

The rest of this paper is structured as follows: Previous work on visual pose estimation and related topics is discussed in the rest of this section. In Section 2 the robotic platform including sensors, hardware and software is described. In Section 3 the visual pose estimation techniques used for data analysis are described. Section 4 presents an analysis of visual pose estimation based on a dataset gathered by the platform. Finally, conclusions and future work are discussed in Section 5.

### 1.1 Related Work

Recently there have been significant advances in the area of real-time 3D pose estimation and scene structure estimation in the area of computer vision. The capabilities to accurately derive the egomotion of single [Nistér, 2004] or multiple [Konolige *et al.*, 2007; Koch *et al.*, 1998;

Pollefeys *et al.*, 2004] moving cameras on robotic platforms has been demonstrated using only visual cues.

#### Pose Estimation on Ground Platforms

Visual pose estimation on ground platforms has been successfully demonstrated by a number of groups with impressive results. Newmann *et al.* have demonstrated visual pose estimation using stereo vision and lasers with loop closure detection of trajectories of several kilometres [Newman *et al.*, 2009]. Konolige *et al.* have also demonstrated vision only pose estimation over several kilometre trajectories over a number of datasets under difficult conditions [Konolige and Agrawal, 2008; Konolige *et al.*, 2007; 2010]. Such robotic pose estimation is well established, but subject to significant drift over large trajectories without external input such as loop closure detection, or additional sensors such as an Inertial Navigation System (INS) and Global Positioning System (GPS) sensor.

#### Bundle Adjustment

Most visual techniques for pose estimation now rely on bundle adjustment as a method of optimising the solution. Bundle adjustment is a now well-established field of computer vision, and a number of differing approaches to the problem are present in the literature [Triggs *et al.*, 2000; Lourakis and Argyros, 2005; Dickscheid *et al.*, 2008; Engels *et al.*, 2006; Lourakis and Argyros, 2009]. Many such schemes make use of GPS or IMU inputs to constrain or assist the motion estimation [Bryson *et al.*, 2009; Clark *et al.*, 2006; Konolige and Agrawal, 2008], but newer methods are capable of high accuracy motion estimation using vision alone [Konolige *et al.*, 2010; Sibley *et al.*, 2009].

The extreme difficulty however, is developing a time-efficient implementation that is both stable, achieves a global minimisation and is capable of real time implementation. Sibley *et al.* and Engels *et al.* present methods for taking advantage of the sparseness of the single camera bundle adjustment problem, and we use similar ideas in our algorithmic solution.

#### SLAM

Konolige *et al.* [Konolige *et al.*, 2007] demonstrated frameSLAM, a visual Simultaneous Localisation and Mapping Method using key frames to reduce the size of the nonlinear system such that the system was highly scalable and computationally efficient. Our work is most closely related to the first core steps taken in frameSLAM: 1) precise, real-time visual odometry for incremental pose estimation, and 2) nonlinear least squares estimation for local registration. Cummins and Newman have demonstrated a number of results that include FABMAP based loop closure and additional inputs to develop a full-SLAM system [Cummins and Newman, 2009]. In our work, however, loop closure is not considered and pose is estimated only from stereo vision.

## 2 Ground Based Platform and Sensor Payload

The platform consists of an off-the-shelf computer system, mission sensors including cameras and GPS, and runs an open source operating system to log all data.

### 2.1 Robotic Hardware

The computer system runs on an Intel Atom Dual Core processor (1.6GHz) on a mini-ITX mainboard. The system contains two 60GB solid state drives in a software based RAID0 configuration to enable the system to log at the required rates. This system allows up to 25 minutes of logging time. The system can be controlled by a single human operator either directly on the system or wirelessly via ssh communication with TPLink wireless 802.11g modules. These allow line of sight communication to a distance of 600m, but in ground based tasks this is significantly reduced.

The system is powered via a 90W DC input voltage (12-25V) power supply, that powers the computer, cameras and GPS module. The cost of the entire system is approximately AUD$2500.

#### Sensors

The sensor payload consists of two firewire 1394B colour Point Grey Flea 2 cameras and a USB NMEA 0183 Global Positioning System (GPS) antenna. The cameras are forward facing in a parallel fashion on the top of the vehicle with an approximately $800mm$ baseline. Images are captured synchronously using timing inputs from the firewire bus in compliance with the firewire protocol. The optical system uses two relatively wide angle $4.5mm$ lenses with a field of view of approximately $59^\circ \times 45^\circ$. This allows a wide field of view that can capture a large scene around the path of the vehicle.

GPS updates are received from a Haicom HI-204 III USB GPS with an approximately $10m$ 2D RMS accuracy which is used as an odometric reference. The GPS receiver is installed in a location directly behind the cameras. The output of this receiver is not used in the visual pose estimation in any way.

### 2.2 Robotic Software

The sensor processing system runs in an Ubuntu 9.10 environment using the Orca Robotics software[1] to interface to sensors and log data.

The cameras acquire Bayer colour coded images at a resolution of $1280 \times 960$ pixels at $30Hz$ over the firewire bus. GPS updates were received at $1Hz$ from the Haicom GPS. The software carefully monitors the images entering the software for lack of synchronisation and resynchronises them before allowing them to be recorded. Debayering of the images is done offline due to the large amount of processing involved in debayering on the CPU, and the rate at which data can be logged to the hard disks. Normal colour images are not passed over the firewire bus due to bandwidth limitations of the 1394B firewire specifications.

[1] http://orca-robotics.sourceforge.net

**Improving logging efficiency**
Logging images at high rates presents a number of computational issues and a minimum number of memory copies is essential. The Orca software framework we use is module based (as are most robotics frameworks). This enables complex systems to be built from simple reusable components. Modules or components must communicate data to each other and this is often achieved through a communication handler (in our case IceStorm) (Fig. 2). This requires multiple deep copies of the communicated

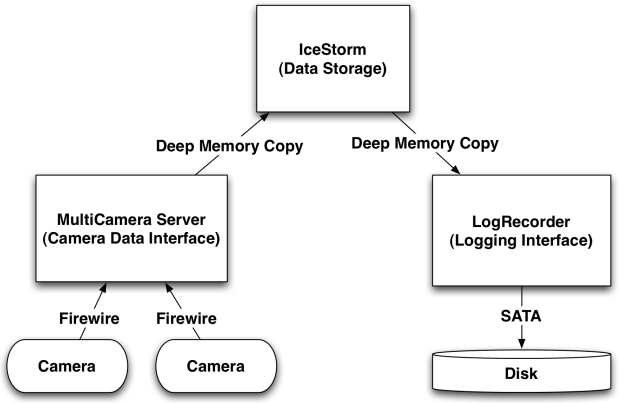


Figure 2: The traditional structure of the Orca Robotics Software for image logging tasks. Modules exist as seperate processes and data is transferred via deep RAM memory copies between each process.

information which is generally computationally inexpensive for most types of data. However, for images at high data rates, this is highly inefficient and is not possible for low speed, low-cost hardware. In order to achieve

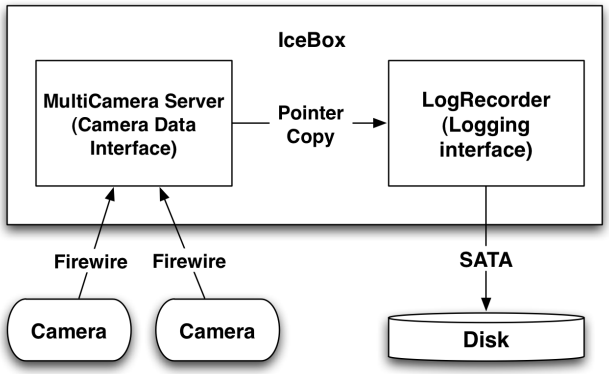


Figure 3: The implementation of the IceBox based structure of the Orca robotics software used in the described sensor platform. Data in memory is no longer deeply copied. Instead, pointers to memory are passed between seperate threads within the single IceBox process.

high data rates, use was made of the Ice middleware (on which Orca is based). Individual components were combined in an Icebox allowing what were originally separate processes to be run as separate threads or services within a larger single process (Fig. 3). Pointers to the data are passed instead of the data itself, and the memory copies only occur once when data is read from the camera. This allows the software to run on much slower hardware without fault.

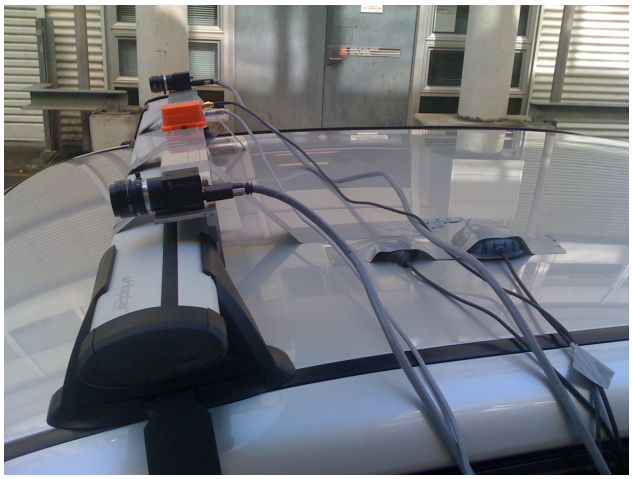

Figure 4: The layout of the sensor apparatus used for experiments

## 2.3 Platform Specific Challenges

There are a number of issues that must be considered when developing any robotic platform in order to obtain high quality data. On this platform, high data rates are essential to log high frame rates at high resolutions. Additionally, vibration and environmental impacts may adversely affect data capture in causing blurred or poor contrast images. Electromagnetic interference and poorly placed wiring may also affect the operation of GPS devices and wireless communication.

**Visual Sensor Issues**
The use of high quality cameras and lenses means that images are of high resolution, in-focus, low distortion and have accurate colour reproduction. Careful consideration must be given to the setup and physical properties of the cameras. The cameras must be able to cope with both bright sunlight and deep shadow, fast movement and high contrast between sky and ground.

For many robotic applications the platform, and subsequently the cameras, are subject to fast motion or high vibration. In certain conditions, such as poor lighting, long shutter times and subsequent motion blur can occur. Any motion blur in an image used for visual pose estimation renders the image useless, and as a consequence a sequence of even two or three frames with motion blur can mean the pose estimate will fail.

To prevent this problem, shutter speed is restricted to $85\mu s$. This ensures that any exposure is short enough to prevent motion blur in almost all circumstances. However, this setting may mean that in low light conditions insufficient light enters the CCD. High gains are therefore necessary to obtain a normally contrasted image but as a consequence increase image noise. It is therefore paramount that the iris is set to a reasonable size to allow sufficient light to enter, and the shutter time is the bare minimum required for the application.

Figure 5: Example stereo pair from the UQ Colleges Dataset showing high image contrast between ground and sky, moving traffic and depth range of potentially trackable features.

**Logging Issues**

The logging system must be able to cope with very high frame rates to ensure significant coverage between frames. Solid state disks have been chosen as the system is intended for use in off-road ground based applications, and inside a small UAV platform. A spinning disk would not perform optimally in the high vibrations of an off-road ground platform or the even more extreme vibration, high temperature and potentially high G environment inside a small aircraft. Instead, solid state disks can perform the job with ease.

In our case we acquire images at an effective rate of 60Hz meaning 45 MB per second must be recorded in a streaming fashion. This is a relatively easy task for a spinning disk but a challenge for solid state drives due to poor write speeds. The poor write speeds are a challenge for solid state disks due to architecture; for any data that is written to a block on an SSD, the block must first be read in order to save the data already on the block, and the whole block rewritten with the new data. In order to counteract the slow write speeds as a consequence of this architecture, two disks were used in a RAID0 architecture to increase the sustained write speed to disk. Additionally, the file system journalling and access time writes are disabled and the scheduling algorithm has been changed to a 'deadline' form to approximate real time operation. Streaming writes are also given extremely high priority while reads are given lowest priority.

**Sensor Calibration**

The most critical aspect of the correct functioning of any stereo pose estimation system is the need for extremely high accuracy calibration of the stereo rig. Intrinsic parameters such as focal lengths, distortions, principle point, and extrinsic properties such as the relative translation and rotation between the cameras must be estimated to within hundredths of a degree.

Fine calibration is required for accurate pose estimation, such that the epipolar geometry of features between cameras must lie within a single pixel row of their intended position. If this is not the case, our algorithm will have difficulty triangulating features and generating accurate pose updates.

Calibration is achieved using a sequence of images containing a checkerboard pattern and the use of the freely available Camera Calibration Toolbox for Matlab[2]. An optimised method of bundle adjustment is performed to accurately determine 3D checkerboard locations and camera pose, while optimising focal length, distortions and camera principle points. Poor images due to low light conditions, motion blur or distortion to the checkerboard are identified and discarded.

For the wide baseline (800mm) required for our vehicle based dataset, at least 80 images of the $0.5 \times 1.0m^2$ checkerboard are required for an accurate calibration. Additionally, the calibration must be either completed with the camera rig in place, or within a short timeframe of the dataset capture due to changes in the physical properties of the rig. Impacts such as high vibration and potential physical bumps are detrimental to any calibrated stereo rig.

The constraint is even stricter in our case due to the wide baseline and potentially highly distant features. Even a slight bump to the rig will cause parameters to change and any previous calibration is rendered useless. In order to counteract these issues the rig is mounted to the vehicle with the ability to allow for vibration and impact dampening as shown in Figure 4.

## 3 Unaided Visual Pose Estimation

Our interest is in using the robotic platform to perform accurate localisation using vision over large trajectories. There are two major tasks to be addressed:

- Using visual data association to give a motion estimate between camera frames
- Optimising the motion estimation by minimising the re-projection error of camera and observed feature positions

The preliminary stage of our pose estimation routine consists of a highly robust Visual Odometry (VO) sys-

[2]http://www.vision.caltech.edu/bouguetj/calib_doc/

tem that simultaneously determines the pose of camera frames and features by matching projected feature positions between frames. The core of our pose estimation routine consists of a multi-camera aware bundle adjustment routine that optimises the VO estimate over a window of up to 12 recent frames.

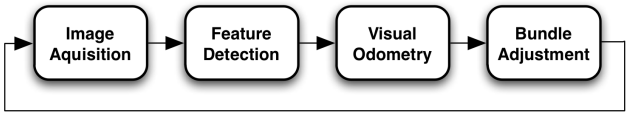


Figure 6: Pipeline of operations in pose estimation algorithm.

### 3.1 Data Association

Generating an incremental VO estimate begins with performing reliable data association between current and previously detected features on each update, in our case this is accomplished by accurately matching image features between frames. Our feature detection routine consists of the detection and generation of SIFT descriptors [Lowe, 1999]. These features are known for their rotation and scale invariance, and accurate matching over wide baselines. Features are matched in a similar method to that described by Lowe. By comparing the Euclidean distance between descriptors and using the ratio of distance between the closest and second-closest match a probability of a match can be established, and those matches below a minimum threshold are rejected.

### 3.2 Motion Estimation

By using the feature matches between frames as a motion model between consecutive frames, an estimate of motion can be generated. This motion estimate can be generated by using as few as three matched features between frames, and we take advantage of this in our method [Haralick *et al.*, 1994]. Feature matches, however, are not strictly reliable, and a set can consist of many false matches.

To actively choose valid matches and produce a robust VO estimate, we use a RANSAC iterative framework [Fischler and Bolles, 1981]. As an additional robustness guarantee, our method only uses features that have been tracked over a minimum of three frames for the VO estimate. This ensures that only stable features that are easily trackable (and hence more likely to be reliable) are used.

**Bundle Adjustment**

While VO can give promising motion estimates, over long distances the estimate of motion can drift significantly. This is often the result of errors in feature observations, poor feature matching and difficulties in tracking certain camera motions such as tilt and roll. We attempt to minimise these errors by passing the VO through a high level bundle adjustment optimisation routine.

Bundle adjustment applies the well known non-linear least squares optimisation method to the large estimation problem of optimising camera frames $c$ and feature positions $p$ to match feature observations $z$ within camera frames. Bundle adjustment is an efficient method of optimisation as it allows a way of expressing frame constraints and uncertainty, and directly associates them with feature measurements. We formulate the problem to use a partitioned Levenberg-Marquardt scheme to reach a global minimum but with a significant number of optimisations.

In essence, bundle adjustment finds the maximum likelihood solution by finding the minimum of the sum of the problem constraints:

$$f(x) = \sum_{ij} \Delta z(x_{ij})^T W_{ij} \Delta z(x_{ij}) \tag{1}$$

That is, what feature observations minimise the camera and feature position estimation errors. The most computationally expensive part of the problem consists of the inversion of the Jacobian at each time step. Our algorithm takes advantage of the sparsity of sections of the Jacobian in the multi-camera case, allowing bundle adjustment to perform quickly and efficiently. In Algorithm 1 we present the pose estimation algorithm in its entirety.

**Algorithm 1** Vision Only Pose Estimation

**Input:** A sequence of high resolution camera frames from two intrinsically and extrinsically calibrated cameras $C_0$ and $C_1$

**Output:** An estimate of camera pose for each frame and 3D point cloud representing visible features

1: **for** all frames in sequence $i = 0, 1, 2, ..., n$ **do**
2: Get frames $C_{0,i}$ and $C_{1,i}$
3: Detect SIFT Features and generate descriptors in frames $C_{0,i}$ and $C_{1,i}$
4: Match features in frame $C_{0,i}$ to previous frame $C_{0,i-1}$
5: Drop all features visible in frame $C_{0,i}$ whose track length (number of appearances in frames before and after) is less than $j$, where $j$ is a user selected window of frames
6: Triangulate features between frames $C_{0,i}$ and $C_{1,i}$ using stereo pixel differences to extrapolate their 3D position
7: Perform an iterative RANSAC based 3 point pose estimate from frame $i-1$ to current frame $i$ using features visible in all four frames: $C_{0,i}$, $C_{1,i}$, $C_{0,i-1}$, $C_{1,i-1}$
8: Perform sliding bundle adjustment window on frames $i-k$ to $i$, where $k$ is a user selected window of frames
9: **end for**

## 4 Results

Our results include a trajectory gathered from the developed robotic sensor system at the University of Queensland's St Lucia Campus. We present a comparison between the final camera based pose estimate and the GPS as a ground truth. Additionally, results are plotted on

a satellite map aligned to the GPS co-ordinate system (Fig. 1).

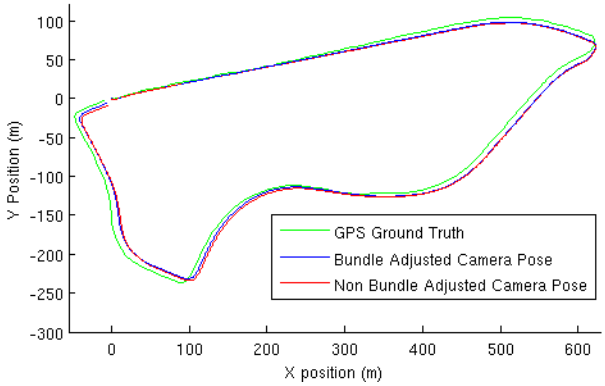


Figure 7: Plotted trajectory of GPS path (green), non-bundle adjusted trajectory (red) and bundle adjusted trajectory (blue).

### 4.1 Dataset Trajectory

The ground platform was driven in an approximately 1.6km loop for a duration of approximately 5 minutes at the St Lucia campus of the University of Queensland, dubbed the 'UQ Colleges Dataset'. It followed established roads on a day with normal traffic movements. The loop consists of a number of sharp turns, traffic obstacles, stop points, variable speeds, speed bumps, lens flare and pedestrian traffic. The weather was overcast, meaning that images were often dark and significant gain noise was introduced in the images. Stereo camera data was logged at a resolution of $1280 \times 960$ pixels per image at 30Hz in a Bayer encoded format. Updates were also received from the GPS unit at a rate of 1Hz for the duration of the dataset.

### 4.2 Pose Estimation Results

We have generated a VO estimate over a section of the collected dataset with only MLESAC based estimation (i.e. without bundle-adjustment) and a VO estimate with multiple-camera bundle-adjustment (in an attempt to assist the trajectory estimation). No loop closure detection or additional sensor input is used to assist the trajectory generation. The output of the visual pose estimation is compared with a GPS ground-truth provided by the NMEA GPS with approximately 250 data points (Fig. 7). Our best pose estimate is over 7000 consecutive stereo frames covering a distance of 1.605km in the bundle adjusted case, and 1.608km in the non-bundle adjusted case, compared to a GPS generated trajectory generated of 1.629km.

The visual pose estimate is generated in a single Euclidean frame with no initial alignment to the GPS co-ordinate system. In order to compare the pose estimate with the GPS ground truth the data is aligned with the GPS pose estimate using an absolute orientation algorithm [Horn, 1987]. By aligning the data, the final pose estimate can be seen in Figure 7 and the error result between camera position and GPS ground truth can be seen (Fig. 8) for both the bundle adjusted and non bundle-adjusted estimates.

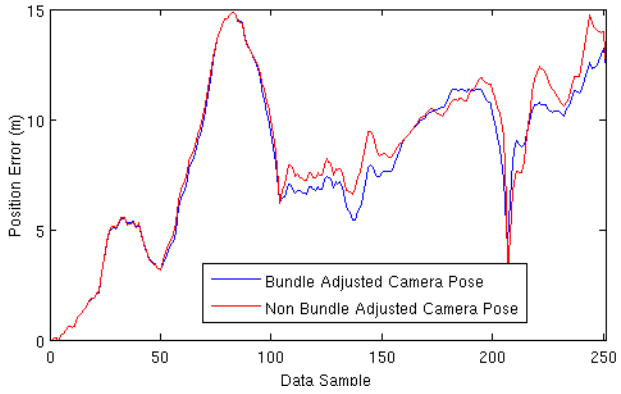


Figure 8: Comparison of error in pose between camera path and GPS in 3 dimensions (X,Y,Z).

**Comparison of Pose Estimate to Ground Truth**
From Figure 8 it can be seen that the final error in pose estimation between the bundle and non-bundle adjusted trajectory is approximately 12.3m for both camera trajectories, with a maximum deviation of approximately 15m. Such high error can be attributed to three factors:

- Poor altitude estimation from the GPS. Output logs suggest that the GPS altitude measurement can drift by up to 10m between two separate readings at the same location.
- Imperfect alignment between camera pose and GPS trajectory via the absolute orientation algorithm.
- Drift in the camera pose estimate. Poor feature tracks and subsequent poor pose estimates will cause the trajectory to drift with increasing distance.

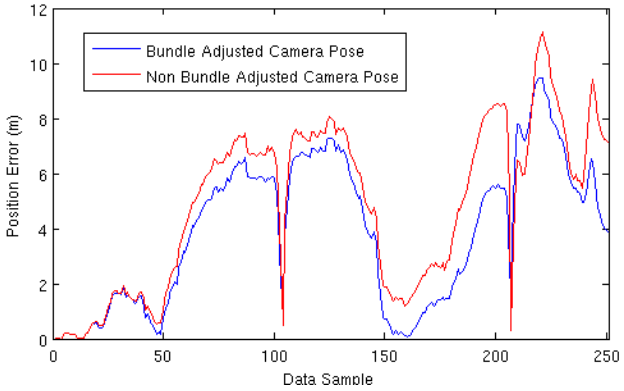


Figure 9: Comparison of minimum squared distance error in pose between camera path and GPS in 2 dimensions (X and Y, neglecting altitude).

Taking into account the first two factors, it is worth considering the geographic 2D comparison while neglecting altitude. As can be seen in Figure 9, this assumption significantly improves the alignment with ground truth, such that the final error in pose compared to GPS for the bundle adjusted case is 3.8m and for the non bundle

adjusted case is 7.1m. The final error in pose can be observed in Figure 10.

The improvement in estimate given by bundle adjustment over the course of the trajectory can also be observed in Figure 9. Bundle adjustment provides a slightly more consistent pose estimate relative to GPS over the length of the trajectory, improving the final position error.

## 5 CONCLUSIONS

We have successfully demonstrated that the sensor platform is capable of logging high quality data for visual motion estimation, and presented highly accurate VO results gathered with the platform.

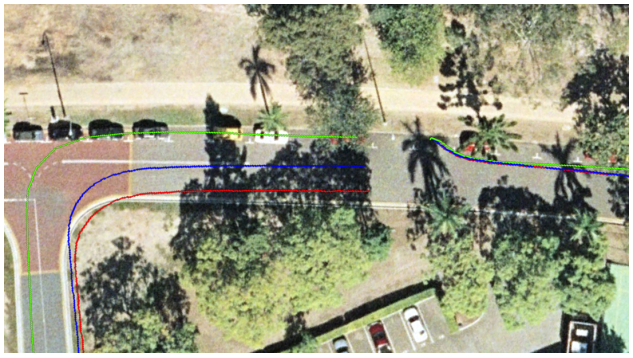

Figure 10: Close up view of final pose estimate at end of trajectory showing GPS path (green), non-bundle adjusted trajectory (red) and bundle adjusted trajectory (blue).

### 5.1 Future Work

There is significant room for feature improvements and additional functionality on our system. For example:

- A high quality Inertial Navigation System (INS) will be installed on the platform to increase the accuracy of ground truth and provide a more quantitative comparison in future datasets.
- The image logging system will be further improved by using advanced image processing techniques to cope with difficult scenes. Methods of handling dynamic gain will be implemented that further improves the robustness of the images to real-world difficulties.
- It is intended that the system will be installed on a small UAV platform using a commercial autopilot for research into airborne vision based mapping. The logging system is small, low-power and high reliability; ensuring that it is a very suitable candidate for installation in a UAV.
- We intend to implement a full-featured metric SLAM system on data gathered from the platform that incorporates position filtering and recognises successful loop closures, improving pose and reconstruction estimates.
- We also intend to implement significant speed improvements to our algorithms by pushing much of the bundle-adjustment and SLAM processing onto low cost graphics processing hardware so that pose estimates can be generated in an online fashion on the platform.

## References


[Bryson *et al.*, 2009] M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *2009 IEEE International Conference on Robotics and Automation*, pages 3143–3148. Ieee, May 2009.

[Clark *et al.*, 2006] R.R. Clark, M.H. Lin, and C.J. Taylor. 3D environment capture from monocular video and inertial data. *Three-dimensional image capture and applications VII: 16-17 January, 2006, San Jose, California, USA*, 2006.

[Cummins and Newman, 2009] Mark Cummins and Paul Newman. Highly scalable appearance-only SLAMFAB-MAP 2.0. In *Proc. Robotics Science and Systems*, pages 1–8, 2009.

[Dickscheid *et al.*, 2008] T. Dickscheid, T. Labe, and W. Förstner. Benchmarking automatic bundle adjustment results. In *XXI. ISPRS congress, Beijing*, pages 7–12, 2008.

[Engels *et al.*, 2006] C Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. *Photogrammetric Computer Vision*, 2, 2006.

[Fischler and Bolles, 1981] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with. *Communications of the ACM*, 24(6), 1981.

[Haralick *et al.*, 1994] BM Haralick, CN Lee, K Ottenberg, and M Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Computer Vision, International Journal of*, 13(3):331–356, 1994.

[Horn, 1987] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629, April 1987.

[Koch *et al.*, 1998] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. *Multi Viewpoint Stereo from Uncalibrated Video Sequences*, pages 55–71. 1998.

[Konolige and Agrawal, 2008] Kurt Konolige and Motilal Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on*, 24(5):1066–1077, 2008.

[Konolige *et al.*, 2007] K. Konolige, M. Agrawal, and J. Sola. Large scale visual odometry for rough terrain. In *Proc. International Symposium on Robotics Research*. Citeseer, 2007.

[Konolige *et al.*, 2010] Kurt Konolige, James Bowman, JD Chen, Patrick Mihelich, Michael Calonder, Vincent Lepetit, and Pascal Fua. View-based maps. *The International Journal of Robotics Research*, 29(8):941–957, 2010.

[Lourakis and Argyros, 2005] M.I.A. Lourakis and A.A. Argyros. Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? *Computer Vision, IEEE International Conference on*, 2:1526–1531, 2005.

[Lourakis and Argyros, 2009] Manolis I. a. Lourakis and Antonis A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Transactions on Mathematical Software*, 36(1):1–30, 2009.

[Lowe, 1999] DG Lowe. Object recognition from local scale-invariant features. *Computer Vision, IEEE International Conference on*, 2:1150–1157, 1999.

[Newman *et al.*, 2009] Paul Newman, Gabe Sibley, Mike Smith, Mark Cummins, Alastair Harrison, Chris Mei, Ingmar Posner, Robbie Shade, Derik Schroeter, L. Murphy, and Others. Navigating, recognizing and describing urban spaces with vision and lasers. *The International Journal of Robotics Research*, 28(11-12):1406, 2009.

[Nistér, 2004] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 756–777, 2004.

[Pollefeys *et al.*, 2004] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[Sibley *et al.*, 2009] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems Conference*, pages 1–8. Citeseer, 2009.

[Triggs *et al.*, 2000] B Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment a modern synthesis. *Vision algorithms: theory and practice*, pages:153–177, 2000.