Chapter 1

Experiments with Real Data

1.1 Equipment Setup and Data Collection

In order to examine the accuracy and feasibility of the algorithm being used in low flying UAV for obstacle detection purpose, realistic aerial video and navigation data are collected through a test flight with the support of Sander Geophysics Ltd. A main purpose of the test flight is to obtain a piece of aerial video with the camera close to the ground as much as possible. This is difficult to achieve with any manned fixed wing aircraft. Therefore, a simulated unmanned aircraft system (SUAS) was used to carry all sensors. The SUAS is then towed by a helicopter via a tow rope of 33 meters long (Figure 1)to complete the survey. Yet, to prevent the SUAS from being caught by tree top, sufficient clearance must be left between the SUAS and the vegetation. As a result, the helicopter flew a planned path at approximately 100 meters above ground, and SUAS at approximately 70 meters above ground.

Sensors mounted on the SUAS included one wide angle CCD camera with 6 mm focal length lens capturing monocular image sequence at 30 fps, a pair of narrow angle CCD cameras for binocular images, one GPS antenna, and one flight control INS/GPS navigation unit Athena GS-111m [1](Figure 2). Analog video and navigation data are sent to the helicopter via three BNC cables and one data cable. Installed in the

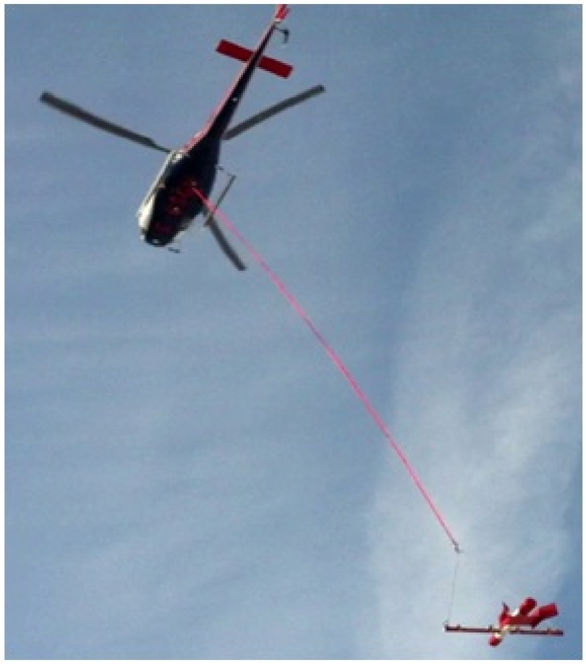

Figure 1: Simulated UAS towed by helicopter

helicopter are two SGL data acquisition system CDAC. This system records video and data from SUAS, as well as data from sensors installed on the helicopter, including GPS, radar and laser altimeter, air pressure, temperature, humility, etc. Navigation data from the SUAS were sent in RS485, and were directly recorded via the UART port of the computer (figure 3). Videos from the three cameras were digitized to 720x480 resolution images using a PC/104+ MPEG4 video encoder from Parvus installed in CDAC. The video were time-stamped with GPS second on the image screen for post-flight synchronization with the navigation measurements. A snapshot of the digitized video is shown in figure 4.

1.2 Camera Calibration

A camera calibration was performed after the flight by taking a piece of video of a checker board pattern with various translation and rotation motion. A total of 20

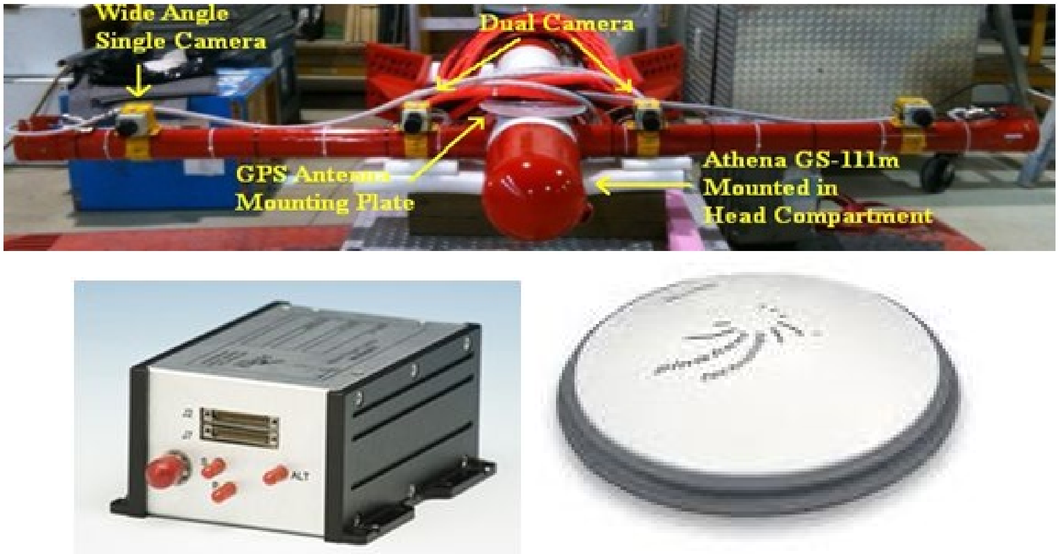

Figure 2: Sensors mounted on SUAS. Top: the SUAS, bottom left: Athena GS-111m, bottom right: GPS antenna

views of the calibration target was chosen from the video, and fed to the calibration algorithm. A few examples were shown in figure **??**. The algorithm,"calibration.exe", used for calibrating the camera comes with the OpenCV installation. Table **??** below listed the calibration results.

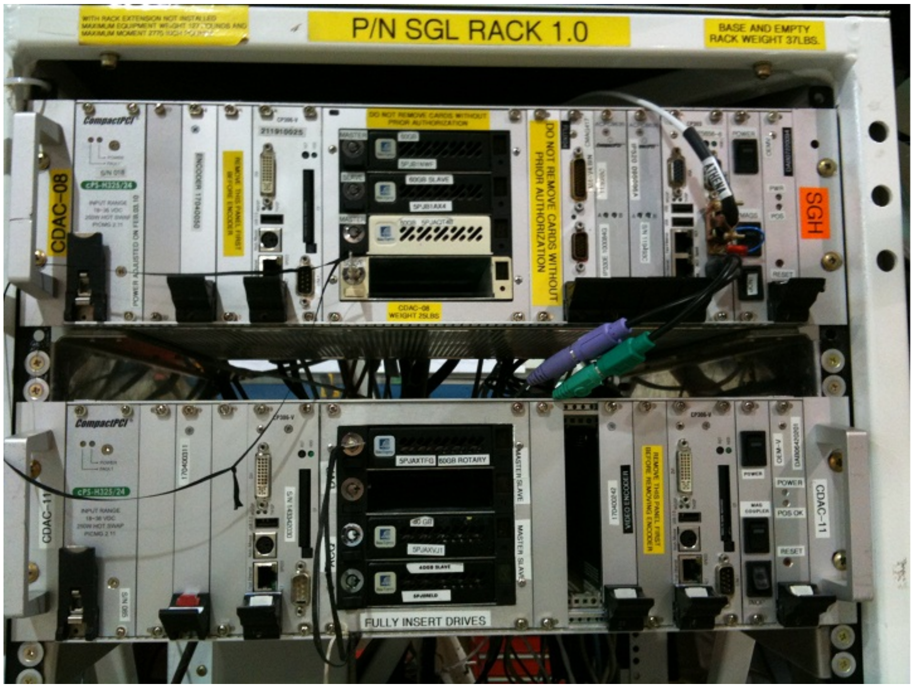

Figure 3: Compact PCI data acquisition system (CDAC)

Image Width	720 pixels
Image height	480 pixels
f_x	887.6
f_y	805.7
c_x	381.8
c_y	293.7
k_1	-0.102
k_2	-0.535
p_1	1.15e-003
p_2	8.40e-003

Figure 4: Image from monocular camera with GPS second timestamp

1.3 Ground Truth Data Collection and Comparison

UAS localization ground truth are obtained through onboard flight control unit GS-111m. The unit records the SUAS position in GPS longitude and latitude coordinate. Orientation is obtained from the roll pitch and heading measurements. Roll and pitch accuracy has 0.1° mean and 0.1° standard deviation. Heading accuracy can achieve 0.5°. [1]

The estimated SUAS position can be directly obtained from the inverse of world origin estimation in the CC_EKF_SLAM state vector: $[O^c_{XYZ}, W^c_{XYZ}]$.

Digital elevation map (DEM) are downloaded from CGIAR-CSI website [2] and used as ground truth data for feature mapping. The downloaded DEM contains longitude, latitude and see level elevation of the terrain with a resolution approximately 100 meters by 100 meters. To compare estimated feature position to the the DEM, some conversion are necessary to bring both data to the same coordinate system. In this work, the comparison are done in UTM coordinate.

First, the longitude and latitude in DEM data are converted into UTM using the WGS84 world geodetic system [3]. Many library are readily available to do the conversion by taking GPS coordinate and zone number as input. The library used in this work is a python interface to PROJ.4 library [4] called pyproj [4]. Secondly, the DEM data in UTM were converted into world frame using transformation matrix with intial SUAS position and orientation as input.

To bring result from the CC_EKF_SLAM algorithm to world frame, feature coordinates must be first converted into world frame using the estimated UAS localization results. Let $[X_i^W, Y_i^W, Z_i^W]^T$ be the feature coordinate in world frame, it can be calculated by

$$\begin{bmatrix} X_i^W \\ Y_i^W \\ Z_i^W \end{bmatrix} = Q^{-1}(O_{XYZ}^c, W_{XYZ}^c) \left(\begin{bmatrix} x_i^C \\ y_i^C \\ z_i^C \end{bmatrix} + \frac{1}{\rho_i} m(\varphi_i^C, \theta_i^C) \right) \tag{1}$$

Appendix A

Coordinate Transformation

For a mobile robot traveling in world. A point in space has coordinate $\begin{bmatrix} x & y & z \end{bmatrix}$ in world frame, and $\begin{bmatrix} x' & y' & z' \end{bmatrix}$ in the mobile frame. The two coordinate is related by

$$\begin{matrix} x \\ y \\ z \\ 1 \end{matrix}_{World} = Q \cdot \begin{matrix} x' \\ y' \\ z' \\ 1 \end{matrix}_{mobile} \tag{2}$$

where Q is the transformation matrix composed by rotation matrix on X, Y, and Z axis and a translation matrix. Q follows the TR?Y convention... (fig)

$Q(r_x, r_y, r_z, T) = Q_{Rz}(r_z) \cdot Q_{Ry}(r_y) \cdot Q_{Rx}(r_x) \cdot Q_T(T)$

$Q^{-1}(r_x, r_y, r_z, T) = Q_T^{-1}(T) \cdot Q_{Rx}^{-1}(r_x) \cdot Q_{Ry}^{-1}(r_y) \cdot Q_{Rz}^{-1}(r_z)$

$$Q_{Rx}(r_x) = [\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & \cos(r_x) & -\sin(r_x) & 0 \\ 0 & \sin(r_x) & \cos(r_x) & 0 \\ 0 & 0 & 0 & 1 \end{matrix}]$$

$$Q_{Ry}(r_y) = [\begin{matrix} \cos(r_y) & 0 & \sin(r_y) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(r_y) & 0 & \cos(r_y) & 0 \\ 0 & 0 & 0 & 1 \end{matrix}]$$

$$Q_{Rz}(r_z) = [\begin{matrix} \cos(r_z) & -\sin(r_z) & 0 & 0 \\ \sin(r_z) & \cos(r_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}]$$

$$Q_T = [\begin{matrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{matrix}]$$

$$Q_{Rx}^{-1} = Q_{Rx}^T$$

$$Q_{Ry}^{-1} = Q_{Ry}^T$$

$$Q_{Rz}^{-1} = Q_{Rz}^{T}$$

$$Q_T^{-1} = Q_T?$$

List of References

[1] "Athena 111m integrated flight control system."

[2] "CGIAR-CSI SRTM 90m DEM digital elevation database."

[3] "World geodetic system - wikipedia, the free encyclopedia."

[4] "pyproj - python interface to PROJ.4 library - google project hosting."