

Traversability Analysis using a UAV mounted LIDAR and camera - Preliminary Report

Paul D. Cox, Intern at LAAS-CNRS, 5AE INSA Toulouse

April 18, 2011

Contents

1	Project Background and Motivation	2
1.1	Report Purpose	2
1.2	LAAS Robotics	2
1.3	LAAS UAVs	2
1.4	ELROB competition	2
2	Project Description	4
2.1	Autopilot System	4
2.1.1	Paparazzi tradeoffs	5
2.2	Airborne Processor	5
2.3	Airborne Sensors	6
2.3.1	Hokuyo UTM-30LX	6
2.3.2	Aptina MT9V032 Camera	6
2.3.3	XSens MTIG INS	6
2.4	Scan Flight Geometry	6
2.5	Challenges	7
2.5.1	Weight and Size	7
2.5.2	Synchronization of Sensors	8
3	Ground-based Testing	9
3.1	Test Description	9
3.2	Output Sample	9
4	Mentor Aircraft Project	11
4.1	Description	11
4.2	Block Diagram	11
4.3	Construction	11
5	Research Perspectives	13
5.1	Traversability Mapping and DTM	13
5.2	Localization and Mapping	13
5.3	UGV UAV cooperation	13

Chapter 1

Project Background and Motivation

1.1 Report Purpose

This report's purpose is to present the ideas behind this project and the progress so far. It also includes practical discussions intended to guide the group as it attempts to broaden its use of UAVs. A literature review has been significantly delayed due to the early technical advancement necessary to adequately plan for the June competition, rendering a theoretical analysis at this stage impossible. The author hopes that by the end of the internship a more than cursory review of the field and how this project fits into it can be presented in the final report.

1.2 LAAS Robotics

The Robotics and Interactions (RIS) group of the LAAS operates a number of robots who serve as testbeds for algorithm development and research [1]. These indoor and outdoor robots are primarily UGVs adapted for various tasks or needs. For example, Rackham (Figure 1.1a) [2] is equipped with a plethora of vision and proximity sensors, while Jido (Figure 1.1b) [3] is outfit with a manipulator, and Dala (Figure 1.1c) [4] is capable of high-speed outdoor travel. The group is interested in complementing its ground-based fleet with airborne vehicles. The intent is to research cooperative schenarios where the airborne vehicles provide a sensing advantage due to their unique perspective and mobility. More specifically, the group decided to include a UAV in this years ELROB competition attempt, and this project [5] will be to create that UAV.

1.3 LAAS UAVs

There have been a number of UAV projects at LAAS: A dirigible named Karma [6], and three fixed-wing aircraft: Lhasa [7], Nirvana, and Manta. The Nirvana aircrafts were used for formation flight [8] and not equipped with a sensing payload. Manta, a flying wing with a payload based on commodity X86 hardware, was abandoned due to it's less than convenient size and EMI issues [9]. These UAVs reuse the autopilot and groundstation software and hardware from ENAC's open-source UAV system named paparazzi [11].

1.4 ELROB competition

The European Land Robot (ELROB) competition is held yearly in Belgium, and the theme alternates every year between military and civilian [12]. For the 2011 event, the competition is divided into four scenarios, namely: Reconnaissance and surveillance, Transport, Camp security, and Autonomous navigation.

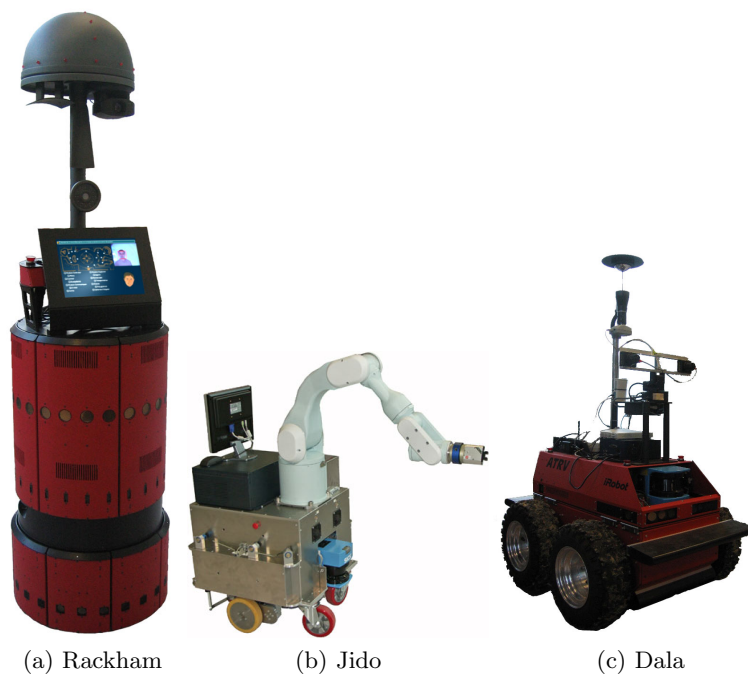


Figure 1.1: LAAS UGVs

Chapter 2

Project Description

The platform selected by the group is a low-power embedded system based on the Texas Instrument OMAP3 processor (details in Section 2.2). The Gumstix Overo module [14], on which the OMAP processor is mounted, provides the capacity to run the Linux operating system with minimal physical dimension and power requirements. The sensors selected include the Hokuyo UTM-30LX (Section 2.3.1), a scanning laser range finder, a CMOS image sensor (Section 2.3.2), and an inertial navigation system (XSens MTIG, Section 2.3.3). Apart from the camera, all these sensors are already used extensively at the LAAS and are supported with custom driver software developed at the LAB [10]. Furthermore, these drivers have already been ported for use with the ARM processor architecture.

2.1 Autopilot System

Paparazzi is an autopilot system based on a C-code mainloop program running on an airborne ARM microcontroller and Ocaml ground-station code running on a standard Linux PC or laptop [11]. The system allows for extensive customization and extensions. For example, the flight plan mechanism allows the creation of complex trajectories based on waypoints and conditions. The system also incorporates various simulation capabilities that are useful in initial testing of new code and configurations. Ground to air communications are typically achieved with low-bandwidth serial modems, support multiple simultaneous aircraft and ground stations, and use a software bus model that makes the creation and insertion of additional software agents into the system straightforward.

Paparazzi supports fixed-wing and rotating-wing aircraft. There is much recent use on quadrotor platforms, especially the Asctec airframe [15], but helicopter use has so far been very limited. Users of the fixed-wing configuration typically use small foam airframes such as the ones manufactured by RC Manufacturer Mutliplex [16], brushless electric motors, and the wingspans remain below 1.5m. In a few cases, the airframes are much larger, or much smaller, or more complex. These might incorporate fiberglass or carbon airframes for extended runtime ?? or range ??, or a gas turbine to achieve higher velocities ??, or larger wing areas and increased payload capacity. Essentially, the PID algorithms implemented in paparazzi are such that just about any standard craft can be controlled if the gains are adjusted properly.

In the most basic and most widely used fixed-wing configuration, the autopilot uses GPS and IR sensors for attitude and navigational control [19]. Due to some of the drawbacks of these sensors, support for replacement or complimentary sensors exists and is continuously explored. This includes the use of static and dynamic pressure sensing (to allow the aircraft to respect air-speed or altitude bounds or setpoints, for example), the use of inertial measurement units (to eliminate the need for a minimum ground/sky IR contrast, for example), and the use of a magnetometer to obtain a measure of the aircraft's heading. To maintain an accurate state estimate with small craft capable of high roll rates, the bandwidth-limited IR sensors are complemented with a roll gyro. This is typically a low-cost MEMS sensor, of the likes from Analog Devices, and interfaced via the autopilot's extra digital interfaces or sampled by an ADC channel.

The current autopilot hardware is the Tiny2.1, a 50-gram board with a NXP LPC2148 microcontroller, a U-blox GPS receiver module and antenna, and small Molex Picoblade connectors for interfacing with the following devices :

UART-based modem (typically Digi XBee/XBee Pro, or Aerocomm) PPM-based RC receiver input

(PPM signal requires receiver modification so certain users have put the effort into using serial-protocol based 2.4GHz receivers) PWM-based RC servo outputs USB for programming by PC Host ADC inputs to sample IR sensors and monitor battery level I2C and SPI interfaces for additional communication, sensor, or actuator options (see examples cited above)

2.1.1 Paparazzi tradeoffs

A typical useful hardware and software system must wrestle with the complexity trade-off. Either designs follow the KISS principle, and favor lightweight and easy to master architectures, or they aim to deliver a high degree of modularity and configurability to enable extensibility, continuous reuse and evolution. Paparazzi adopts the KISS principle for hardware designs, choosing to use COTS modules and skipping any redundancy, subscribing to the idea that a simple, straightforward, clean, and well-documented design can be relatively robust and enables a larger community (for many, a hobby activity). The software side is very different because extensibility and configurability were goals from the outset and development has continuously evolved without any milestones or official releases. This has results in a complex and sparsely documented code base that can seem unwieldy to anyone except a dedicated developer with the proper amount of pre-requisite competencies.

Like all active complex configurable systems, the challenge with paparazzi is its steep learning curve and rapid and continuous evolution. Tailoring the system for a specific application is typically not difficult because paparazzi requires extensive modifications. Instead, it is difficult because much of the system has to be mastered before a clear picture of how and where the modifications should be made. Since documentation is limited to unstructured wiki pages and due to a complete lack in code comments, a user is essentially forced to be a developer and a developer is forced to dive deep into the system or resort to hacks to limit the time invested. Beyond the obvious drawbacks of the later option, is the facts are only useful for the very short-term. Since the hacks will not be able to be merged with the software base, as the base evolves the hacks will continuously break and require constant effort to keep them working. This pressure is the reason why many users have forked the paparazzi code, added their extensive application-specific modifications without staying up to date, to end up with a system that is nearly unmergeable with the evolved paparazzi code base. The bottom-line is that tailoring paparazzi for a project should be approached in one of two ways depending on the objectives. One, if the intent is to produce a one-off demo and future reuse is not a consideration, a fork can be considered along with quick hacks. If the continuous evolution of the application is envisioned or if sharing the capabilities with other paparazzi users is desired, the time to properly integrate customizations and continuously maintain them needs to be anticipated and allocated.

All this to say that considerable time is required integrate paparazzi in this project and that new team members will not able to contribute for the few months it will take to get to know the system.

2.2 Airborne Processor

The central component of the payload is the Gumstix Overo ARM processor board. This TI OMAP System On Chip (SOC) processor incorporates a 600MHz 32-bit ARM core, a DSP, and many peripherals including UART, I2C, SPI, SD, USB, wifi, bluetooth, and a dedicated camera interface. While the camera interface and the DSP provide for future research, for this project the important interface is USB Host as this will be used to receive laser scanner telemetry from the Hokuyo device along with other serial-based communications (MTIG and ground communication).

The OMAP processor runs a full Linux operating system, cross compiled on a standard Linux host using a compilation system called OpenEmbedded. OE is capable of generating the necessary bootloader, linux kernel and modules, along with a root file system populated with all of the libraries and applications that might be needed to run our payload code. Configuration is done using bitbake recipes. With assistance from LAAS developers, bitbake recipes were created for the hokuyo and MTIG device libraries and are now available for the OMAP.

2.3 Airborne Sensors

2.3.1 Hokuyo UTM-30LX

The UTM-30LX scans a single line around it's central axis at a rate of 40 Hz. A distance and reflectivity measurement is taken at a resolution of 1/4 of a degree, it's field of vision is 270 degrees, and distance resolution is in the millimeter range. The USB interface, along with open-source software drivers [10], allows easy interfacing with PC and embedded hosts.

2.3.2 Aptina MT9V032 Camera

The MT9V032 CMOS device from Aptina was selected due to its global shutter characteristic that guarantees that all pixels in a frame are acquired simultaneously (unlike the much more common rolling shutter cameras found in most webcams, cell phones, etc). Camera resolution is 752 x 480 pixels and is available in color or black and white.

The CMOS Camera is interfaced to the OMAP processor using the dedicated camera bus by means of level converters (1.8V/3.3V). This direct connection is a parallel bus with a maximum clock rate of 40MHz, which allows for fast and efficient acquisition of image data. A custom designed camera board using this sensor is in the works at the LAAS while initial testing will depend on the commercially available camera board from Gumstix, INC 2.3.2.

Software support is provided as a linux driver, currently for kernel version 2.6.34. The driver has been tested to work and performance remains to be determined.

2.3.3 XSens MTIG INS

The XSens MTIG device uses MEMS sensors along with a GPS receiver to estimate attitude and position estimates at a maximum rate of 100Hz using an undocumented Kalman filter running on its embedded processor. The interface uses USB and the serial protocol is well documented and supported with various open-source libraries including ??.

2.4 Scan Flight Geometry

nominal UAV flight velocity : 20-30 m/s nominal UAV flight height AGL : 30 m Lidar sensor resolution : 1080 points over 270 deg visible (1440 points over 360 deg) @40Hz

ground covered distance during one revolution of scanner:

$$Dist_{per_scan_rev} = ground_speed \times time_{per_scan_rev} = 20 \frac{m}{s} \times \frac{1}{40} s = 0.5 m \quad (2.1)$$

For 90 °; interest zone :

scan line advances down ground track :

$$Dist_x = \frac{90}{360} \times Dist_{per_scan_rev} = \frac{1}{4} \times 0.5 m = 12.5 cm \quad (2.2)$$

scan line proceeds along sensor rotation (for a 90 scan, this is twice the AGL height) :

$$Dist_y = 2 \times AGL = 2 \times 30 m = 60 m \quad (2.3)$$

Resolution :

$$\frac{\frac{90}{360} \times 1440 pixels}{scan_length} = \frac{360 pixels}{\sqrt{Dist_x^2 + Dist_y^2}} \approx \frac{360 pixels}{Dist_y} = 6 \frac{pixels}{m} = 17 < /math > cmbetweenpixels \quad (2.4)$$

Angle relative to track :

$$Angle_{scan_to_track} = \tan^{-1} \frac{Dist_x}{Dist_y} = \tan^{-1} \frac{0.125}{60} = 0.119^\circ < /math > (negligiblerelativetocrabangle) \quad (2.5)$$

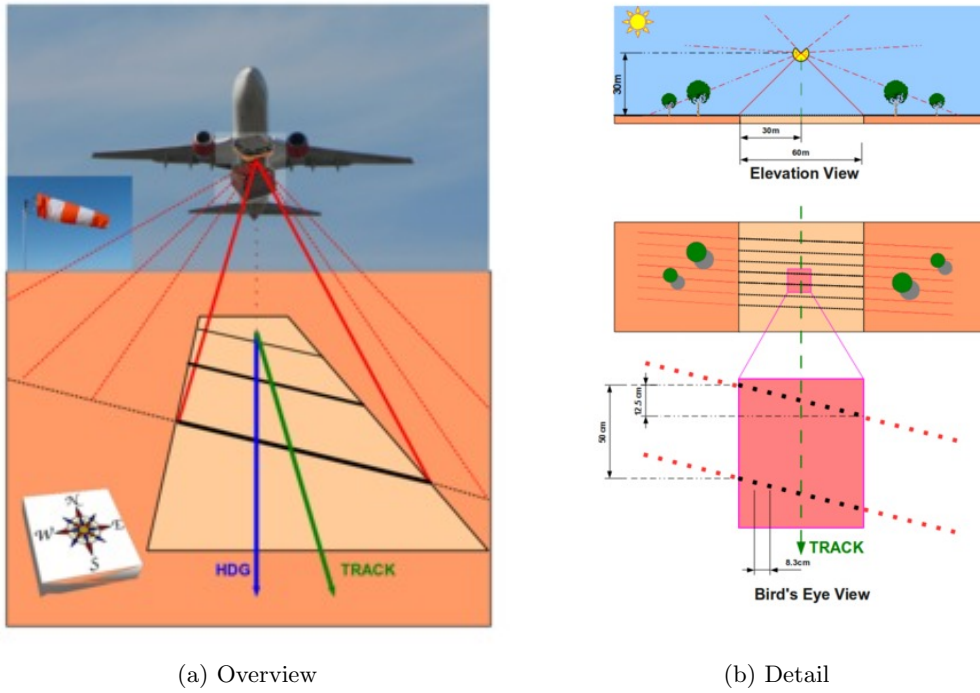


Figure 2.1: Geometry

2.5 Challenges

2.5.1 Weight and Size

The most convenient UAVs are those in the 1 meter wingspan range and made of foam, as these are easy to transport, repair, and pose very little security risk. But their limited payload capabilities (300 grams or less) restrict their practical uses to payload that can be highly miniaturized. In our case, where a payload capacity of 600 grams is necessary, staying in this 1-meter regime would be possible but requires the design of a custom aircraft. With the mentor airframe, transport and repair options are still manageable, but the high-speed necessary to generate the required lift (anticipated 30+ mps), combined with a 2.5 kg overall weight, also combined with the low altitude necessary for lidar acquisition, means that a crash is more likely due to lower reaction times and proximity to obstacles, and the increased momentum will have more grave consequences. The solution is an airframe and wing design tailored for low-speed flight. This reduces risk and increases the density of the lidar-acquired point clouds. The down-side is the aircraft will have a reduced capacity to fly in stronger winds and a reduced range. The design of this aircraft would include a number of benefits over the current craft:

- a payload bay integrated in the fuselage for reduced drag and increased protection in a crash
- Increase in wing area to permit slow flight
- Reduced wing span for easy of transport
- Airfoil design optimized our needs
- : Difficult to stall and always stays close to the designed airspeed*

(When the aircraft attempts to gain speed in a dive, the airfoil generates additional drag making the accumulation of airspeed difficult.)

Particularly in a lab setting, the major difference with the development and deployment of UAVs is that a generic kitchen sink approach is no longer possible without a large support team and an immense budget. When the research emphasis is on algorithm development, the most useful and practical research UGVs are the ones that resemble a desktop development environment (x86 PC, for example), can house a plethora of sensing and communications devices, and can be easily reconfigured. Due to these goals, UGVs are typically 5 to 100 kilograms. In the UAV context, 5 kg is already a very substantial aircraft. The conclusion is that expectations for small lab UAVs should be that each can only be good for exploring a few ideas after which it will either need to be abandoned or completely reworked. And since it's just not possible to halt an aircraft at a moments notice, crashes are more likely and it should be expected to have to rebuild periodically or

build multiple aircraft up-front to avoid long reconstruction delays.

2.5.2 Synchronization of Sensors

Currently there is loose synchronization based on timestamps added to the scan and INS datastreams at the application level. Tighter synchronization is planned in the future by time stamping at a lower level (i.e. in the device drivers) and by hardware sync between the scanner and the camera. The pulse output generated by the scanner at the start of each revolution will be used to trigger the exposure of a camera frame using its exposure input. This input will be available on the custom camera design currently in development at the LAAS.

Chapter 3

Ground-based Testing

In an attempt to gain some practical experience and flesh out any limitations of the hokuyo device in an outside daylight environment, a ground-based test was carried out.

3.1 Test Description



Figure 3.1: Bike Test Vehicle

The hokuyo sensor along with the MTIG was mounted on boom attached to a standard bicycle. While the bicycle traveled through a residential neighborhood a small laptop acquired the data to files. These files show that on a sunny day the hokuyo can detect at a light-colored perpendicular surface up to 20 meters away. It remains to be determined what distances can be expected on surfaces such as dirt and grass and at angles approaching 30 to 45 degrees from perpendicular, as will be the case for our UAV.

3.2 Output Sample

Here we present a set of screenshots showing the animations produced from the bike test data.

A composite image is generated including a number of text data such as the MTIG output angles and a graphical representations of laser scan data, the location data, and video taken by a boom mounted digital camera. The hokuyo data was recorded at a rate of approximately 20 Hz while the MTIG Data was near 100Hz. Rough synchronization is accomplished by using the MTIG data sample with a timestamp closest to the one in a laser scan (each laser scan is also timestamped).

The next step is to represent the laser scans in a 3D environment, either to create a Digital Terrain Model or to show in a tool such as GDHE. This process is accomplished with the LAAS tools robotpkg and Jafar, and I am still trying to learn these so this task is not yet complete. To become accustomed to these tools and understand the methods behind them, I've developed various applications to work through data

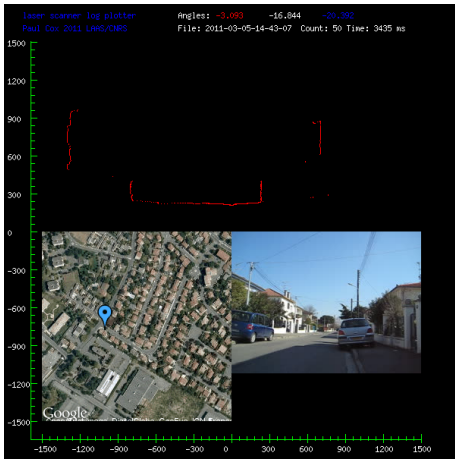


Figure 3.2: Sample animation frame

acquisition, generation, and processing. The code is available freely online [?]. Some screenshots of these applications are below.

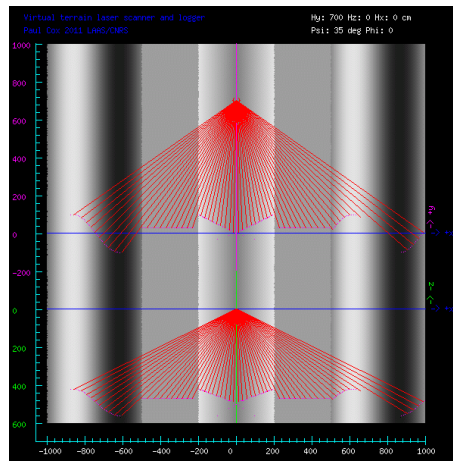


Figure 3.3: Virtual terrain simulator

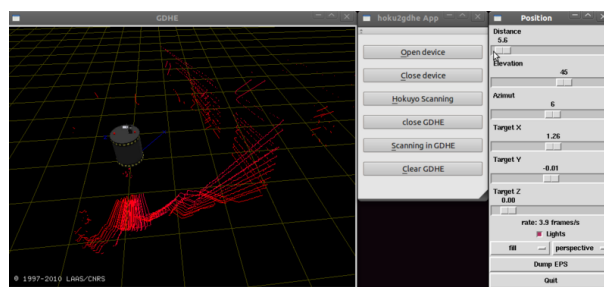


Figure 3.4: Data visualisation in GDHE

Chapter 4

Mentor Aircraft Project

As explained above, the immediate goal is to outfit a Multiplex Mentor aircraft with a Hokuyo, Gumstix Overo, and MTIG for the June 2011 competition. Towards this goal, I've integrated a paparazzi autopilot in a standard Mentor airframe and built a reinforced modular payload pod.

4.1 Description

The aircraft is a standard three-axis electric R/C model. It uses two ailerons, an elevator and rudder for control along with a 40A speed controller for thrust control.

The components are listed below :

insert list here

4.2 Block Diagram

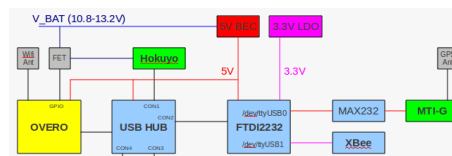


Figure 4.1: Payload Hardware Block Diagram

4.3 Construction

The pod is built using typical model construction techniques. A plywood frame houses the various components and a glass-fiber reinforced foam cover provides a streamlined shape and protects the pod contents in the event of a crash. The payload weight is 650 grams and will require the Mentor to fly at high speed to generate the necessary lift.

Insert aircraft and payload weight tables here.

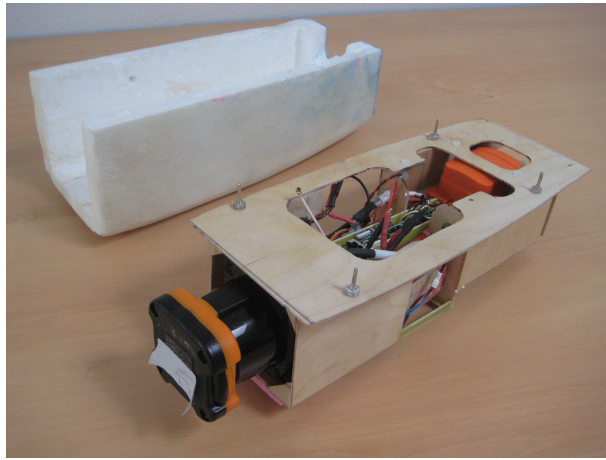


Figure 4.2: Completed Payload Pod



Figure 4.3: Completed Mentor Aircraft with Payload

Chapter 5

Research Perspectives

For the time being, focus is on the practical realization of a useful data set and the research aspect is limited to asking general questions about potential goals and techniques.

Much research literature on applications of airborne lidars : very common in forestry, land management, archaeology.

Techniques to reconstruct ground level by filtering obstructions, or to model buildings or land parcels, determine tree heights or diameters, estimate shore lines.

Much research uses complementarity of vision with lidar.

visual servoing?

5.1 Traversability Mapping and DTM

Constructing DTM, determining traversability from DTM, types of traversability maps... Obstacle detection using aerial Lidar and/or vision. Does the use of topographic maps known a-priori change the nature of the problem? since paths can be determined from the topo maps, UAV tasks is essentially to detect obstacles along those paths.

5.2 Localization and Mapping

Not directly pertinent, but useful to understand how a UGV acquires and models his environment in order to determine how best to assist with UAV options.

5.3 UGV UAV cooperation

UAV by default scans in area complementary to the area under acquisition by UGV. An area beyond and slightly overlapping with the UGV field of vision. Need to determine in what scenarios UGVs would request specific areas to be scanned. For example, if robot is at a split in a path, request to UAV to map both paths. roads.

Since the data volume is large, determining what info needs to be processed onboard and what is transmitted on ground is crucial. The tradeoff is that compute power on ground is easy but getting the data to the ground is not. Processing with simplified algorithms risks filtering useful information so a very specific goal needs to be defined and enough analysis performed to determine that . Also transmission techniques with lossy links is also a large field of study in and of itself.

Bibliography

- [1] <http://homepages.laas.fr/matthieu/robots/>
- [2] <http://homepages.laas.fr/sara/robots/rackham/index.php>
- [3] <http://homepages.laas.fr/matthieu/robots/jido.shtml>
- [4] <http://homepages.laas.fr/sara/robots/dala/index.php>
- [5] <http://paparazzi.enac.fr/w/index.php?title=Laserhawk>
- [6] <http://homepages.laas.fr/matthieu/robots/karma.shtml>
- [7] <http://homepages.laas.fr/sara/robots/lhassa/index.php>
- [8] Gautier Hattenberger, *Formation flight: evaluation of autonomous configuration control algorithms*. homepages.laas.fr/simon/publis/HATTENBERGER-IAV-2007.pdf Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference, 2007.
- [9] <http://homepages.laas.fr/bvandepo/wiki/doku.php?id=descriptionmanta>
- [10] <http://www.openrobots.org/wiki>
- [11] http://paparazzi.enac.fr/wiki/Main_Page
- [12] <http://www.elrob.org/>
- [13] processors.wiki.ti.com/index.php/OMAP3_Overview
- [14] http://www.gumstix.com/store/catalog/product_info.php?products_id=211
- [15] <http://www.asctec.de/home-en>
- [16] <http://www.multiplexusa.com/>
- [17] Murat Bronz, Jean Marc Moschetta, Pascal Brisset, Michel Gorraz *Towards a Long Endurance MAV* http://www.recherche.enac.fr/LOTA/lib/exe/fetch.php?id=pascal_brisset&cache=cache&media=03-bronz_moschetta_ijmav.pdf www.emav09.org/EMAV-final-papers/paper_78.pdf International Journal of Micro Air Vehicles Volume 1 · Number 4 · December 2009
- [18] <http://paparazzi.enac.fr/wiki/Corsica>
- [19] Pascal Brisset, Antoine Drouin, Michel Gorraz, Pierre-Selim Huard and Jeremy Tyler *The Paparazzi Solution* http://www.recherche.enac.fr/paparazzi/papers_2006/mav06_paparazzi.pdf MAV '06, October 2006
Juchler, Norman, Homography-based Method To Generate A Traversability Map Using Aerial Images, LAAS-CNRS Master Thesis, April 27, 2009
E. Andersen and C. Taylor. An automatic system for creating geo-referenced mosaics from mav video. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, pages 1248-1253, 2008.

- F. C. Benitez. Homography based Techniques for Unmanned Aerial Vehicle Self-Localization using Monocular Imagery. PHD thesis, 2007.
- S. Bosch. Contribution a la modelisation d'environnements par vision monoculaire dans un contexte de robotique aero-terrestre. PhD Thesis, 2007.
- Bosch, Sebastien, Lacroix, Simon and Caballero, Fernando. Autonomous Detection of Safe Landing Areas for a UAV from Monocular Images. 2006
- E. Malis, F. Chaumette, and S. Boudet. 2-1/2-d visual servoing. IEEE Transactions On Robotics And Automation, 15(2):238-250, 1999.
- E. Malis and E. Marchand. Experiments with robust estimation techniques in real-time robot vision. Technical report, 2006.
- E. Malis and M. Vargas. Deeper understanding of the homography decomposition for vision-based control. Technical Report 6303, INRIA, September 2007.
- N. Vandapel, R. R. Donamukkala, and M. Heber. Quality assessment of traversability maps from aerial lidar data for an unmanned ground vehicle. IEEE Int'l Conf Robots and Systems IROS 2003, pages 305-310, October 2003.