On-board computational efficiency in real time UAV embedded terrain reconstruction

Panagiotis Partsinevelos[1], Ioannis Agadakos[2], Vasilis Athanasiou[2], Ioannis Papaefstathiou[2], Stylianos Mertikas[1], Sarantis Kyritsis[2], Achilles Tripolitsiotis[1], Panagiotis Zervos[1]
[1]*School of Mineral Resources Engineering, Technical University of Crete, Chania, Greece, Email: ppartsinevelos@isc.tuc.gr*
[2]*School of Electronics & Computer Engineering, Technical University of Crete, Chania, Greece*

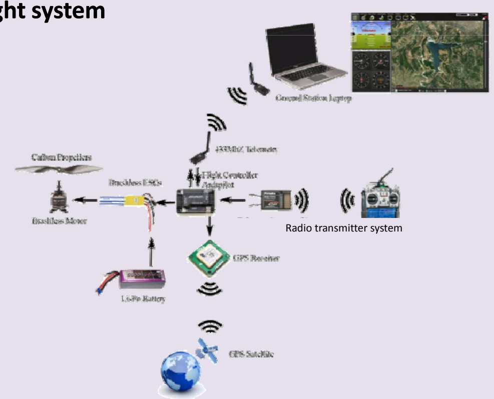
Introduction

In the last few years there is a surge in unmanned aerial vehicle (UAV) related applications aiding recognition, interpretation and mapping. Their specifications are highly diverse with contradictory characteristics including cost-efficiency, carrying weight, flight time, mapping precision, real time processing capabilities, etc.

In this study, a hexacopter system is employed for near real time terrain mapping. The main challenge addressed is to retain a low cost system with real time processing capabilities. The weight limitation affecting the overall flight time, makes the selection of processing components particularly critical. On the other hand, surface reconstruction as a computational demanding task urges for a highly demanding processing unit. To merge these two contradicting aspects along with customized development, a System on a Chip (SoC) system is proposed as a low-power, low-cost processor which natively supports camera sensors and GPS systems.

Modern SoCs such as Omap3530 or Zynq, are classified as heterogeneous systems and provide a versatile platform allowing access to both general purpose processors such as the ARM11, and specialized processors such as a digital signal processor and floating field-programmable gate array. A UAV equipped with the proposed embedded platforms, allows on-board terrain reconstruction using stereo vision in near real time. Furthermore, according to the frame rate required, additional image processing may concurrently take place, such as image rectification and object detection.

Flight system



UAV

Custom made hexa-copter with gross payload of 2 kilos and flight time of 15 minutes

Equipment

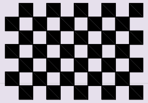
APM 2.6 (Arduino Embedded system as a Hardware Component)
Overo Gumstix with DSP enabled configuration
2x Logitech USB webcams

Software and firmware:
Mission planner 2.0 (compiled for linux)
ArduCopter 3.0.1 (APM 2.6 copter firmware)
Custom Gumstix OS image
video4linux
C algorithm for Calibration, Depth Map and 3D Reconstruction, OpenCV Library

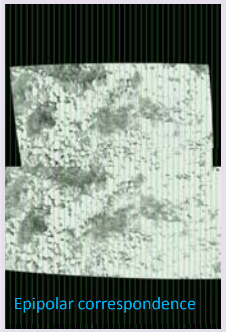
Processing

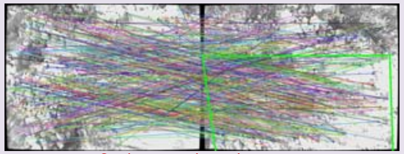
A. Single Camera Calibration
Calibrate each camera in order to generate its intrinsic parameters (focal length, principal point). Create the the camera matrix, distortion coefficients and the fundamental matrix.


Initial image pair


Calibration target

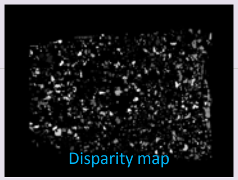
B. Match points and compute homography
Feature/template point matching between image pairs: Accept matches that fall onto the corresponding epipolar lines. In order to check this condition, the fundamental matrix must be known, and initial "good" matches are required to estimate the matrix.
Homography matrix includes extrinsic parameters (translation rotation matrices)

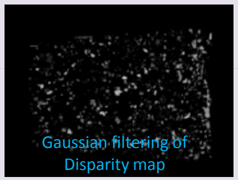

Epipolar correspondence


Conjugate point pairs

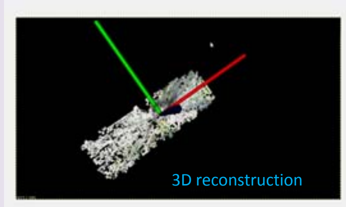
C. Rectification
Rectification takes into account the calibration information of the individual cameras and aligning of two images and extracts rotation matrix and reprojection matrix, that is disparity-to-depth mapping matrix.

D. Disparity computation
A template feature matching algorithm based on sum of absolute differences computes the disparity map for pixels of the image aided by the epilolar geometry.


Disparity map


Gaussian filtering of Disparity map

E. Depth map generation
Given the diparity and the Q reprojection function generated on rectification step, we calculate the 3d coordinates. The Depth Map gives us the distance of each point from the pinhole camera.


3D reconstruction

Real time performance discussion

- Disparity was computed in real time through the use of the computational efficiency of SoC systems with parallel processing capabilities.
- In order to achieve real time performance on non-specialized hardware, such as FPGA's or ASICS, we need to exploit the fact that each pixel on grayscale images is a 2^8 value. On some modern systems such as the selected OMAP3530 it is possible to utilize on the DSP subsystems up to 16 operations per cycle for short integers up to 2^8 bits. By exploiting this subsystems capability and the fact that this co-processor runs without an Operating System we can achieve real time performance when calculating the correspondence window. Each operation is independently executed to each element, thus taking full advantage of the DSP subsystem capabilities.
- Furthermore, by limiting the maximum disparity to a maximum value allows us to limit the search space greatly. For instance on a VGA image setting the maximum disparity value of 64, cuts the search space on each line by a factor of 10 (each line has 640 pixels and we only consider 64 as potential matches).
- Since the whole application runs on the DSP subsystem, the main processor is free to perform additional operations such as feature detection or fleet operations rising the efficiency of the entire system.

Conclusions & future work

- Through this preliminary study, it is shown that the proposed system provides partially real time efficiency,
- Real time refers to disparity map generation (fraction of a second). The acquisition rate was about a second per image,
- Calibration is performed once per camera, and thus is not required to be performed in real time,
- Image loading, rectification, and 3D reconstruction need about 2-3 seconds for an image pair. In the future, by using SoC parallel processing these operations will be performed in real time,
- The image quality affects greatly the final 3-D map. Our results suffered in precision and resolution mainly due to the low resolution, unstable USB web cameras used,
- Nevertheless, there is much potential in on-board computational efficiency under time constrains, and further experimentation will take place,
- For applications such as those encountered in GIS systems, by utilizing known control points, we will be able to better decide between potential disparity planes caused by information loss due to various reasons, e.g. occlusions.