Visual Odometry Onboard a Micro Air Vehicle Using Snapdragon Flight

Eric Solomon[*], Cyrus Vorwald[†], Vikram Hrishikeshavan[‡],
Inderjit Chopra [§]
Alfred Gessow Rotorcraft Center, Department of Aerospace Engineering,
University of Maryland, College Park, MD 20742

ABSTRACT

Visual odometry has been demonstrated with success in drones through onboard flight computers. However, these methods often suffer from significant drift and are limited by the computational power of the onboard processors. In 2014, the Snapdragon Flight was released by Qualcomm as a robust flight computing platform. It couples fast, low-powered processors with a well-featured sensor array and a variety of options for usage with peripherals. This paper details a method of using the Snapdragon Flight in parallel with a smaller flight computer to create an extensible flight computing platform. In this process, visual inertial odometry is implemented on the Snapdragon Flight and is experimentally verified. The results show that simple uni- and bi-axial position tracking is feasible. Data of more complex scenarios is tested, but is not shown, as the results are not well understood at this point.

*

Nomenclature

ABI	Application binary interface
API	Application program interface
ARM	Advanced RISC Machine
DSP	Digital signals processor
ELKA	Enhanced Lightweight Kinematic Autopilot
ESC	Electronic Speed Controller
Hexagon	Snapdragon 801 digital signals processor
IDL	Interface Description Language
IPC	Inter-process communication
Krait	Snapdragon 801 apps processor
mAH	milliAmp hours
OpenCV	Open Computer Vision
PX4	Open source flight stack
QAIC	Qualcomm Interface Compiler
QURT	Qualcomm Real-time Operating System
RAM	Random Access Memory
ROS	Robot Operating System
SDK	Software development kit
SoC	System on Chip
UART	Universal asynchronous receiver/transmitter

INTRODUCTION

There has been growing interest in use of small scale multicopters and helicopters to conduct autonomous operations for a wide range of missions such as: navigation through small openings (windows, pipes), disaster scenarios involving collapsed buildings and logistical support through delivery of items to unknown locations. In order to accomplish these, information relating the position and orientation to an outside environment is a major requirement for autonomous operation of unmanned air vehicles (UAVs). To estimate relative position and orientation in potentially GPS-denied environments, the UAV needs to know information about its environment. At the same time, the UAV needs to have an estimate of its pose in order to describe the environment. Major breakthroughs to solve this problem of simultaneous localization and mapping (SLAM) have occurred since the turn of the century (Refs. 1, 2). The ability to describe an environment based on image data allows for accurate position and orientation estimation. However, image data consists of very large sets of pixel information that can be computationally expensive to analyze. Powerful computers with large sensors were capable of providing accurate information to solve this problem in real-time. Because of the complexity of camera-based localization, computation was generally carried out using an offline computer.

In recent years, due to the rapid advancement in microelectronics, the focus has shifted to autonomous localization on systems with minimal processing speed and smaller, less accurate sensors. There has been widespread research on using vision aided algorithms to help in object tracking and odometry on mobile devices (Refs. 3–7). It has therefore become possible to perform simultaneous localization and mapping possible in real-time onboard a quadrotor drone and has already been attempted by researchers (Refs. 7, 8) (Fig. 1).

However, many of these research algorithms have been developed with proprietary collaboration with industry and are therefore not open to other research groups. As a result, UAV autonomy research has generally been based on cus-

[*]Graduate Research Assistant, esolomon1221@gmail.com
[†]Underraduate Research Assistant, vgt085@gmail.com
[‡]Assistant Research Scientist, vikramh@umd.edu
[§]Alfred Gessow Professor and Distinguished University Professor, chopra@umd.edu
7th AHS Technical Meeting on VTOL Unmanned Aircraft Systems and Autonomy, Jan 24-26, 2017, Mesa, AZ

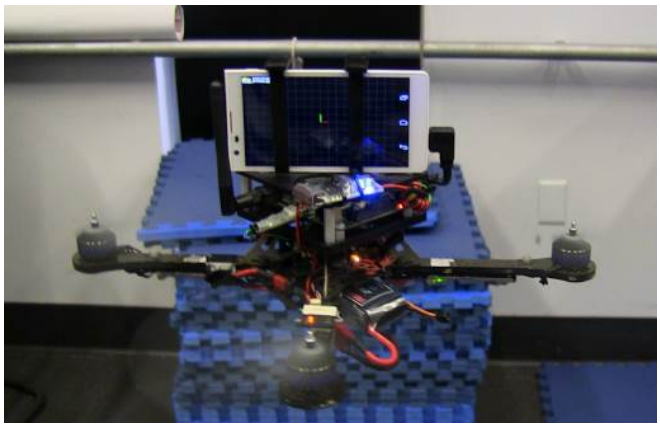

Fig. 1. Visual odometry onboard a 500 gram quadrotor using Android device (Ref. 8)

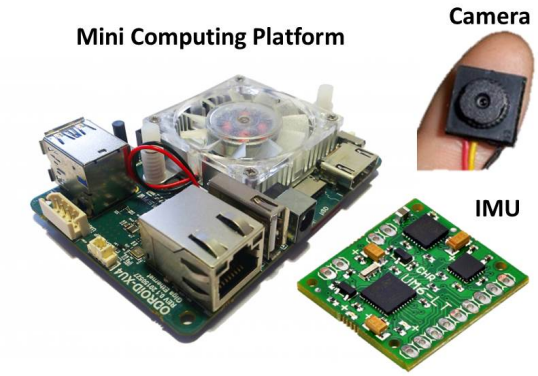

Fig. 2. Typical off-the-shelf components used for custom solution for autonomy onboard a UAV - not weight and power optimized

tom hardware and open source algorithms and development tools (Refs. 12, 13, 16, 29). As can be seen from Fig. 2, the weight allocation for these components is sub-optimal for incorporation with a size and weight constrained micro air vehicle. In order to optimize the hardware choice with computing and power requirements, Qualcomm has released a low-cost ($400) hardware development platform - the Snapdragon Flight (Ref. 34). This has all the hardware and software components required to develop a complete vision based autonomous drone. However the development of this platform is still at a nascent stage and currently can only be used with Qualcomm's custom electronic speed controller board.

It is therefore very desirable to improve upon the functionality and versatility of this development platform. Therefore, the purpose of this paper is two fold: (1) describe the hardware architecture of the Snapdragon Flight and its implementation in a resource constrained localization system that is capable of being used onboard a micro air vehicle, and (2) enhance the class of drones that can be operated with the Snapdragon Flight by integrating it with a custom embedded lightweight kinematic autopilot (ELKA) (Ref. 11) that was developed at the University of Maryland. In this manner, the vision tasks are performed by the Snapdragon Flight and all vehicle flight control tasks are performed by ELKA (Figure 3).

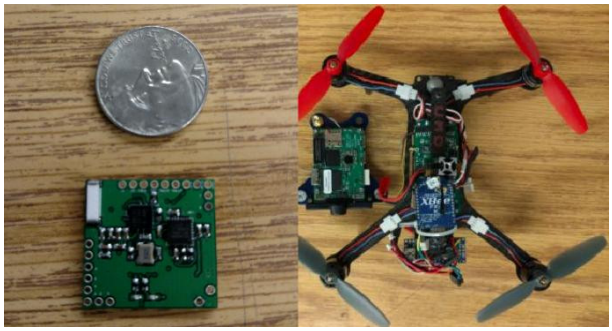

Fig. 3. (a) ELKA and nickel coin (b) Snapdragon Flight, quadrotor drone.

BACKGROUND AND METHODOLOGY

As mentioned before, the purpose is to realize a localization hardware solution that can be applied to mini drones. Fig. 4 shows the hardware architecture of a typical mini-quadrotor with onboard localization. ELKA controls the attitude of the drone at a rate of 1000 Hz. It receives inputs about the drone position from the Snapdragon Flight board at an expected rate of about 20 Hz. This information can then be interpreted towards position feedback control and path planning purposes. The end result of this platform is that the drone software is not tightly coupled to a specific drone hardware.

Next, we discuss the usage of the Snapdragon Flight, often referred to as the Flight, as a feasible drone platform for completing visual inertial odometry. The Flight platform is introduced in terms of hardware and firmware. This paper details the Flight's limitations of communication as well as a proposed solution. Then, the paper discusses visual inertial odometry; first by introducing the general state of visual inertial odometry today as it applies to small onboard computers, and followed by a description of visual inertial odometry as it fits into our specific drone schematic. The paper ends with the procedure and discussion of a simple experiment, which demonstrates the capabilities of the Flight for onboard visual inertial odometry.

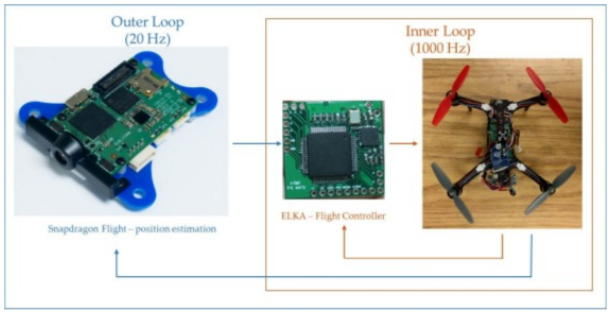

Fig. 4. Hardware architecture

SNAPDRAGON FLIGHT DESCRIPTION

Hardware

The Flight uses the Snapdragon 801 System on Chip composed of a quad-core Krait 400 processor, a Hexagon QDSP6v5 digital signal processor, and an Adreno 330 GPU.

The Krait processor is similar to an ARM Cortex-A processor. The Hexagon DSP is low-power, and efficiently built. It is capable of simulating three independent computing cores, which are useful for parallel code execution. Qualcomm does not make the Adreno 300 series GPU readily accessible to outside developers, so it is thus not used in the project (Ref. 34).

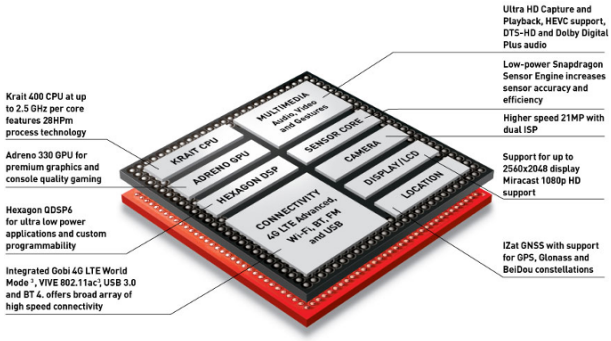

Fig. 5. Snapdragon 801 processor (Ref. 30).

The Flight board has a mass of 26g and consumes 5V DC power through a 2 to 6 series cell (2S-6S) external battery. The input voltage is regulated to 5V through a power adapter. The Flight contains 2GB random access memory (RAM) and 32GB of embedded MultiMediaCard (eMMC) persistent memory. The forward facing camera can capture 4k video at 30 frames per second. The downward facing camera, which performs optic flow velocity sensing, can capture 1080p video. Optic flow analyzes information from consecutive images to estimate vehicle velocity. The Flight has access to 2G/5G WiFi, Bluetooth, and GPS radio communication. Users may attach larger antennae to each radio communication module if desired. Input/output (I/O) is available through USB 3.0, micro SD, serial board connection, gimbal connection, and electronic speed controller (ESC) connections. Digital I/O may be done through protocols including Universal Asynchronous Receiver/Transmitter (UART), Inter-integrated Circuit (I2C), Serial Peripheral Interface (SPI). A 9-axis inertial measurement unit (IMU) may be accessed through SPI, and a barometric pressure sensor may be accessed through I2C (Ref. 34)(Fig. 6).

Firmware

The Flight's Krait and Hexagon processors utilize efficient communication methods to maximize parallelism. The Krait runs QRLinux, a best-effort scheduling operating system that includes Ubuntu 13.04 pre-built binaries (Ref. 24). The Hexagon DSP runs Qualcomm Real-Time OS (QURT), which

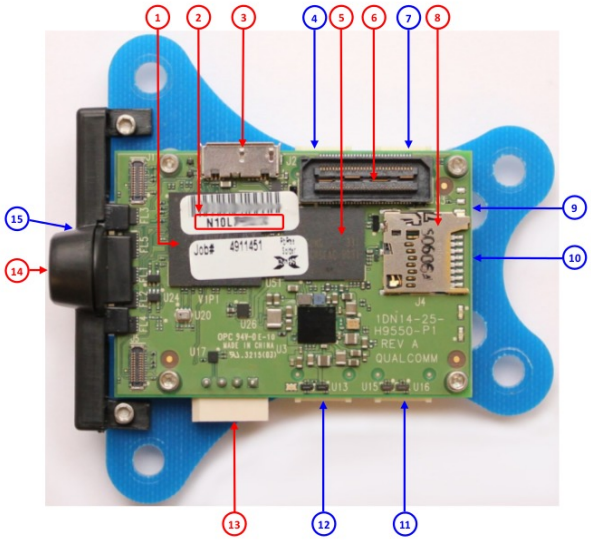

Fig. 6. Description of Flight - (1.) RAM pop memory on top of Snapdragon 801 SoC, (2.) Serial number, (3.) Micro USB 3.0 (Type AB)(OTG)port, (4.) Remote receiver connector, (5.) eMMC flash, (6.) Serial board connector, (7.) IMU connector, (8.) Micro SD card slot, (9.) GPS antenna connector, (10.) WLAN antenna connectors, (11.) Gimbal connector, (12.) ESC connector, (13.) Power connector, (14.) 4K camera, (15.) Optic flow camera (Ref. 28)

is able to efficiently map software processes onto hardware threads without resource overlap. The Hexagon DSP has access to all of the devices attached to the Snapdragon 801 SoC. The the Hexagon DSP is used to access devices and delegate tasks based on hardware efficiency and software support. Visual odometry occurs on the Krait processor, while serial communication tasks occur on the Hexagon DSP. Fig. 7 shows a simple diagram of how different necessary portions of this system are accessed.

Serial communication

As shown in Fig 8, serial communication extends from the Hexagon DSP. Serial communication is a device stream available as '/dev/tty-1...6'. These streams may perform UART communication using either two or four wires, depending on their configuration. Here, two-wire communication is used. The communication packet structure is fairly simple for this program, but is subject to change as the needs of the program change. It consists of a start byte, a control byte, up to 253 data bytes, then an end byte. The packet loading protocol in the program is as yet undefined, but it may be prudent to use a ring buffer data structure to load packets. This will ensure that writing to the packet loading queues will never result in blocking (unnecessary pauses in program) and that only sufficiently recent packets will be sent via UART to ELKA.

VISUAL INERTIAL ODOMETRY

Using computer vision, it is possible to perform simultaneous localization and mapping in real-time onboard a UAV. The

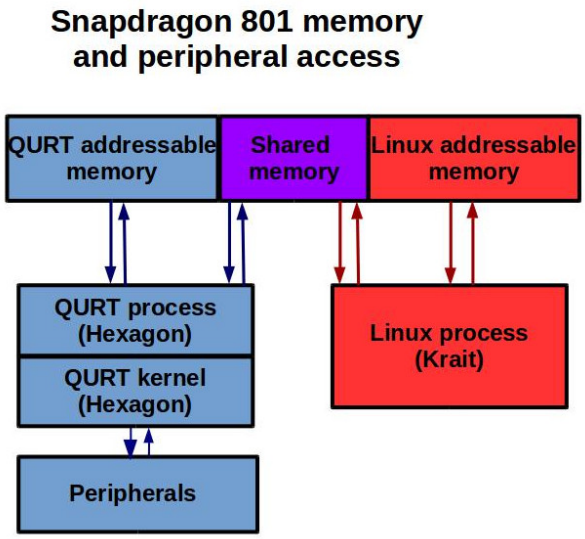

Fig. 7. The division of addressable memory and corresponding addressing processors for the Snapdragon 801. Peripherals are accessed via the Hexagon DSP.

ability to describe an environment based on image data allows for accurate position and orientation estimation. Because of the complexity of camera-based localization, computation is generally carried out using an offboard computer (Ref. 10). Refer to the 'Results and Discussion' section to see that the Snapdragon Flight is capable of localizing using onboard sensors and online processing.

Simultaneous Localization and Mapping (SLAM)

A typical onboard SLAM approach consists of bundle adjustment using a monocular camera and scalar repositioning using an IMU (Ref. 16). The camera frame is a rigid body driven by translation and rotation parameters, as well as estimates of translational and angular velocity. Bundle adjustment requires strong feature correspondence across keyframes with an initial estimate of keyframe poses. The IMU frame measures linear acceleration and angular rates, aiding in the estimation of feature location and keyframe poses. Sensor fusion is typically performed with an error-state Kalman Filter to smooth out the trajectory estimated by the vision algorithm with the IMU (Refs. 15, 36). A working solution that can be used to fuse gyroscope, accelerometer and pose measurements is described in (Ref. 27). The experiment in this paper uses Qualcomm's Machine Vision SDK, for which the source code is not publicly available, in order to show the feasibility of a resource constrained localization system that is capable of being used onboard a micro air vehicle.

Software Architecture

The Krait processor runs a Robotics Operating System (ROS) network to handle multi-processing and asynchronous mes-

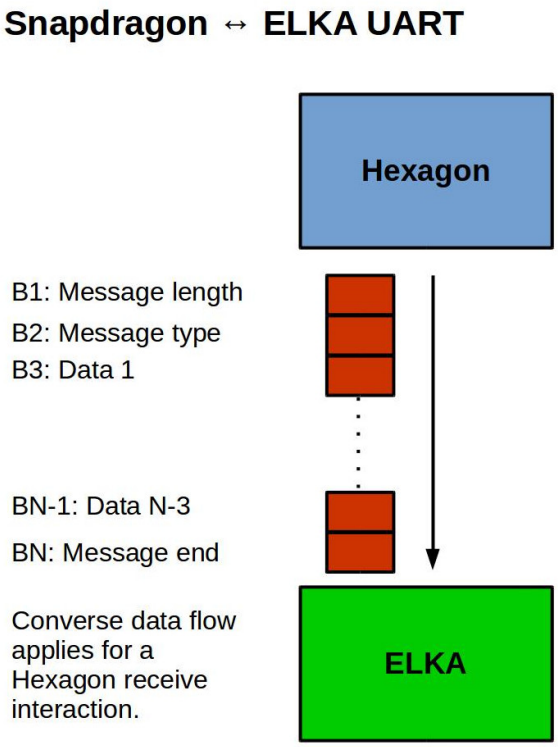

Fig. 8. Snapdragon and ELKA send UART messages back and forth. The format (bytewise) is {start,control,data1,...,datan,end}. Implicity, the max message length is 256 bytes.

saging. ROS is the industry standard for middleware open-source communication between robotic systems. ROS nodes perform three important functions: visual odometry, inter-process communication, and access to additional sensors attached to the Flight. The software control layout is shown in Fig. 9. The visual odometry node uses Qualcomm's Machine Vision SDK. ROS interfaces with PX4, an open-source flight stack, through a single ROS node. PX4 on the Krait processor relays between ROS and Hexagon DSP processes. PX4 on the Hexagon DSP relays between Krait processes, the Flights sensors, and the serial driver. The serial driver relays between PX4 and ELKA, communicating via 2- wire UART. ELKA runs FreeRTOS, which is a simple, multi- threaded real-time operating system. It is capable of receiving motor commands and sending stabilizing control signals to electronic speed controllers attached to motors.

Experimental Measurements

Qualcomm's Visual-Inertial Simultaneous Localization and Mapping package (VISlam) is used for the Flight's onboard tracking and Vicon is used offboard as ground truth data for comparison. Vicon is a motion capture system that consists of cameras with IR light emitters that detect markers that reflect the IR light. The Vicon-460 camera is shown by Windolf, Gotzen, Morlock to be accurate on average between $65-129\mu m$ using four cameras. The setup used in this test

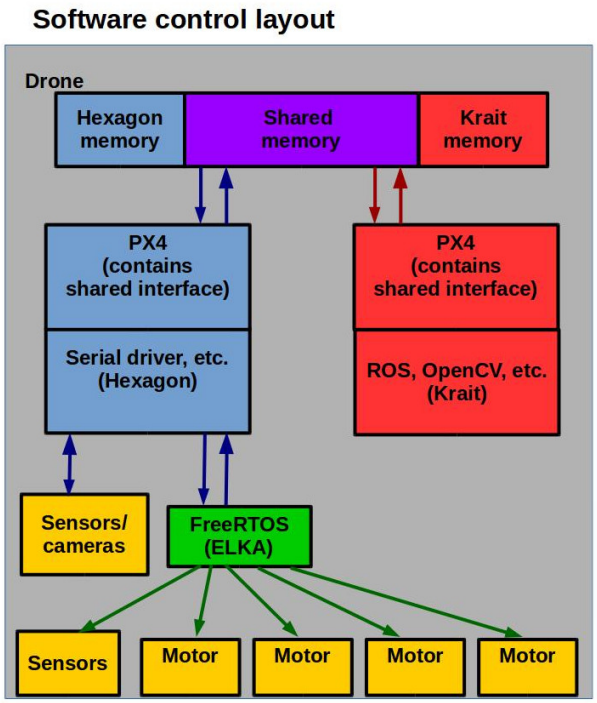

Fig. 9. Software control layout containing the major processes of each processor: ROS, OpenCV, PX4, Serial driver, ELKA.

has seven cameras, although not all of them will track each marker at all times (Ref. 37).

VISlam capabilities for odometry are tested in four characteristic motions: along a single axis, along two axes, along two axes with in-plane rotation, and in arbitrary motion. For the first three test types, the Flight is attached to a rod mounted to a cart, which is rolled around reference lines on the ground. For the third tracking test, the Flight is detached from the cart and moved around at arm's length (Fig. 10, 11). This is the least conservative case for the Flight tracking actual drone motion. This case is not shown, as the results are not well understood at this point.

The Flight is controlled via serial access with Android Debug Bridge (ADB) or via WiFi with Secure Shell (SSH). Flight tracking data due to VISlam is captured onboard. Vicon data is captured on a desktop computer.

Results and Discussion

We present an investigation into the usability of a visual-inertial system that estimates 6 degrees of freedom (DOF) pose in unknown environments. The Flight's output pose ran at an average of 25.189 hertz, which is sufficient for real-time. Results are plotted in the section entitled 'Visual Inertial Odometry Test Data'.

Fig. 12- 14 show position data along simple uni-axial test paths. Fig. 15, 16 show position data along bi-axial test paths. It is shown that the onboard visual inertial odometry results agree well with the Vicon ground truth data. All results show

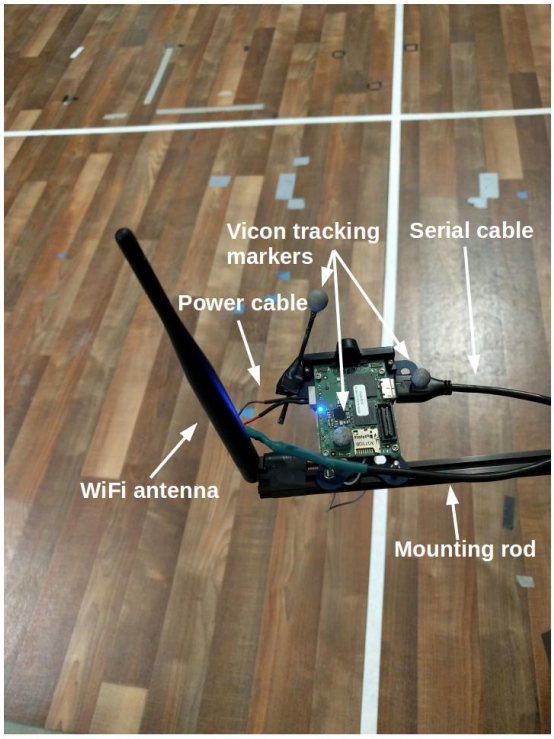

Fig. 10. Snapdragon flight mounted on a rod showing WiFi antenna, power cable, serial cable, and Vicon tracking markers.

a small initial deviation between VISlam and Vicon test data, which does not increase for the duration of the tests. Note that the graph orientations of VISlam data sets are slightly skewed due to the difference between the known Vicon axes orientations and the unknown VISlam axes orientations.

The tests shown do not comprehensively show the Flight's capabilities for visual inertial odometry. The Flight is tracked for approximately 3m in 10s with at most one turn. They are, however, an encouraging starting point and show enormous potential for tracking a complex environment for onboard autonomous navigation.

CONCLUSIONS

The Snapdragon Flight is a strong platform for consideration of onboard autonomous navigation. Its lightweight sensor array supports state-of-the-art navigation techniques. The results show that the Snapdragon Flight is capable of successfully tracking its location relative to a starting point in real-time with low error.

Future work

Currently, UART loopback has not been successfully tested on the Snapdragon Flight. However, the techniques used for

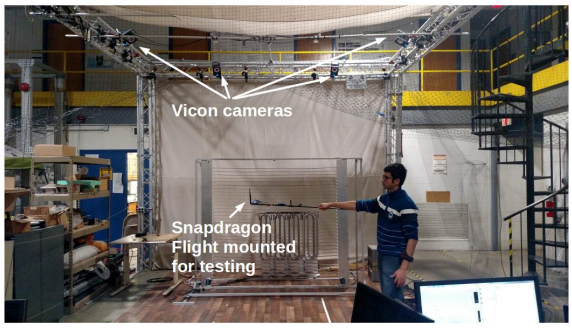

Fig. 11. Visual inertial odometry test setup showing Snapdragon Flight mounted to rod for testing. Tracking due to the Vicon cameras was compared with tracking due to onboard visual inertial odometry.

UART have been successful on similar embedded systems. Possible this discrepancy arises as a result of peculiarities related to the Flight, such as the serial driver or kernel implementations. These problems will probably dissipate assumming that surrounding community support and internal support by Qualcomm for the Flight continues remains in an active state.

After these problems are resolved, the Snapdragon Flight will be set up to show preliminary capabilities controlling a drone during flight. The outputs of VISlam will be fed into ELKA to simply demonstrate the capabilities of the two of them combined for flight.

This system will fly drones such as the AirEZ developed at the University of Maryland by Hrishekeshavan, Phillips, Rand, Chopra (Ref. 17).

In parallel, visual inertial odometry will be verified and extended by using custom implementations of open-source algorithms for the Flight. This way important internal metrics of visual inertial odometry can be quantitatively measured for a variety of algorithms: including feature detection, loop closure, sensor drift, and pose estimation rate. Doing so will also allow the open-source community to test experimental algorithms onboard a very current and relatively high powered onboard flight computing platform.

VISUAL INERTIAL ODOMETRY TEST DATA

All axes referenced are defined by the Vicon camera system. This uses forward-left-up axes as defined from the perspective in Fig 11 for this experiment. All positions are displayed in meters from the center of the Vicon test area.

Linear Movement Tracking

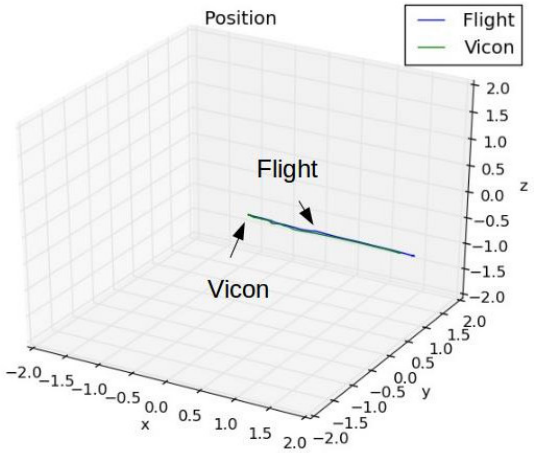

Fig. 12. Positive x position tracking.

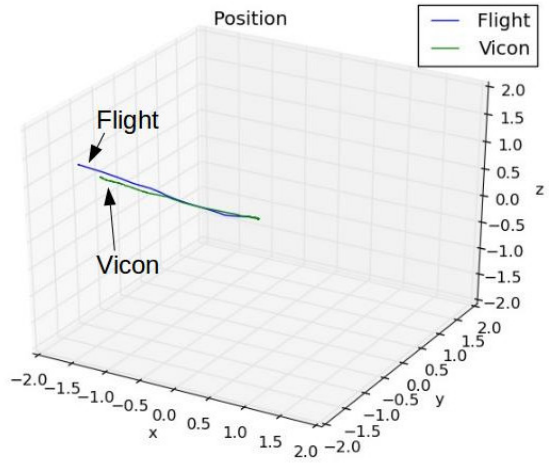

Fig. 13. Negative x position tracking.

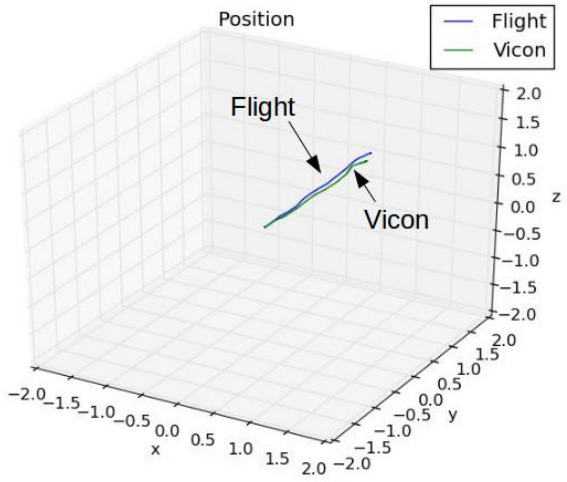

Fig. 14. Negative y position tracking.

Bi-directional Movement Tracking

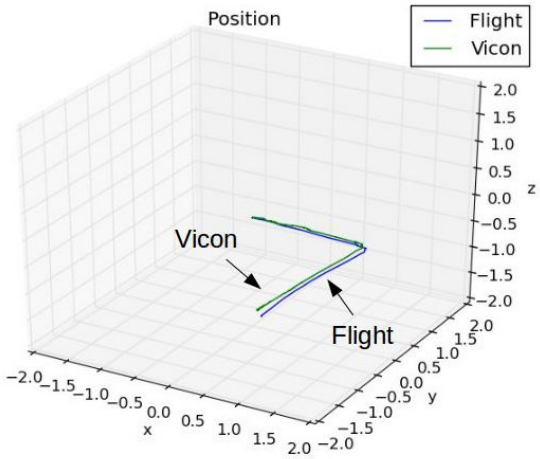

Fig. 15. Positive x, positive y position tracking.

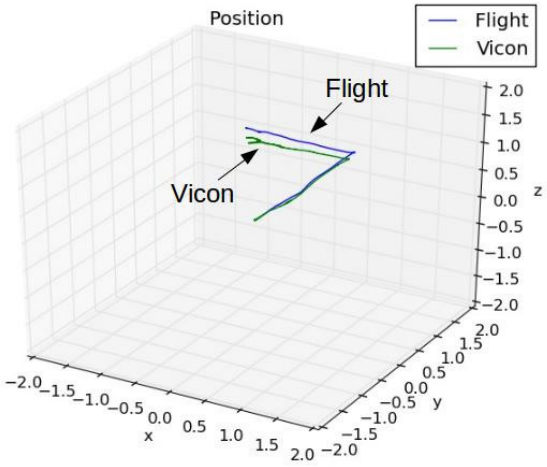

Fig. 16. Negative x, negative y position tracking.

APPENDIX

APPENDIX-1: INTER-PROCESS COMMUNICATION

The Hexagon DSP and the Krait processor handle inter- process communication (IPC) by shared memory. On the Snapdragon 801 SoC, the user controls shared memory through the Qualcomm Interface Compiler Interface Defini- tion Language (QAIC IDL). With the QAIC IDL, the user can declare function prototypes that can be called from the Krait processor and then defined and executed on the Hexagon DSP. The QAIC then performs two major actions: it creates the ap- propriate function stubs (essentially function definitions) for each processor, and it creates libraries that map the interface functions to shared memory. This way, function calls are not different to the user versus how they normally appear, with the exception that only the functions defined in the interface can be used to communicate between processors (Fig 17).

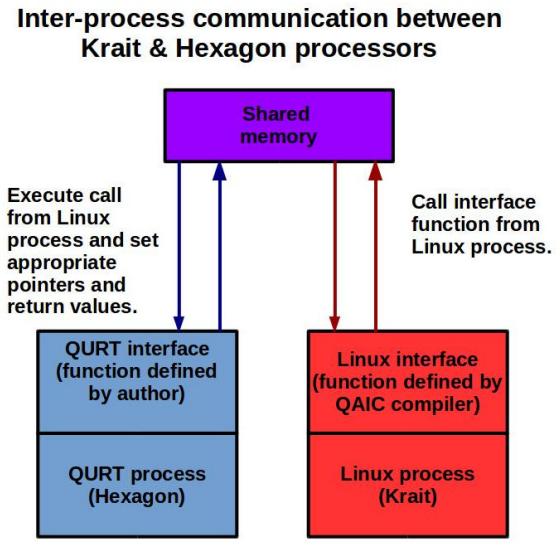

Fig. 17. Snapdragon 801 uses two types of processors almost entirely independently. They communicate with each other thru an inter-process communication scheme defined by the QAIC IDL.

APPENDIX-2: ISSUES AND SUPPORT

Refer to the Qualcomm developer forums (Ref. 7), Intrinsyc support support.intrinsyc.com, ROS forums, and the PX4 discussion forums (Ref. 8) for support on any issues that are encountered dealing with these platforms. Refer to the PX4 github.com/px4and ATLFlight github.com/atlflight Github pages and the ROS wiki (Ref. 9) for updates on program development and issues. In the following table is a list of useful tools for programming with the Flight.

Table 1. Useful tools for programming with the Flight

Name	Use	Reference
PX4	Open soure flight stack for Snapdragon Flight and various other drone platforms	dev.px4.io
ROS	Robot operating system for Ubuntu Linux flavors and some derivatives	wiki.ros.org
Hexagon SDK & Hexagon Tools	Developer kit for programming with Hexagon DSP (includes documentation)	developer.qualcomm.com/software/hexagon-dsp-sdk
Intrinsyc support	Support and documentation for Intrinsyc/Qualcomm products	support.intrinsyc.com
Machine Vision SDK	Machine vision for Snapdragon Flight	developer.qualcomm.com/hardware/snapdragon-flight
CMake	Automated Makefile generation	cmake.org
cmake_hexagon	Cmake build tools for Snapdragon Flight	github.com/atlflight/cmake_hexagon
DSPAL	Hexagon Digital Signal Processor Abstraction Layer	github.com/atlflight/dspal

REFERENCES

[1]Klein, G., and Murray,D., "Parallel tracking and mapping for small AR workspaces," Proc. of the International Symposium on Mixed and Augmented Reality , Nara, Japan, No. 2007.

[2]Whyte, H.D., and Bailey, T., "Simultaneous localization and mapping: Part I," *Robotics and Automation Magazine*, Vol. 13, No. 2, pp.99110, 2006.

[3]Mourikis, A.I., and Roumeliotis.S.I.,"A multi-state constraint kalman filter for visionaided inertial navigation," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 35653572, Roma, Italy, April 2007.

[4]Li,M., Kim,B.H., and Mourikis,A.I., "Real-time motion estimation on a cellphone using inertial sensing and a rolling-shutter camera," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 46974704, Karlsruhe, Germany, May 2013.

[5]Engel. J., Sturm, J., and Cremers, D., "Semi-dense visual odometry for a monocular camera," IEEE International Conference on Computer Vision, pp. 14491456, Dec 2013.

[6]Engel. J., Sturm, J., and Cremers, D., "Scale-Aware Navigation of a Low Cost Quadrocopter with a Monocular Camera," *Robotics and Autonomous Systems* (RAS), volume 62, 2014.

[7]Weiss, S., Achtelik, M. W., Lynen, S., Achtelik, M. C., Kneip, L., Chli, M., and Siegwart, R., " Monocular Vision for Longterm Micro Aerial Vehicle State Estimation: A Compendium,' *Journal of Field Robotics*, 30(5), 803-831. 2013

[8]Loianno, G., and Kumar, V., "Smart phones and flying robots,", Robotics Science and Systems Conference, Berkeley, USA, July 2014.

[9]200qx drone frame, `www.horizonhobby.com/product/multirotor/multirotor-aircraft/bind-n-fly-15087--1/200-qx-bnf-with-safe-technology-blh7780`, 2017.

[10]Engel J., Sturm J., and Cremers D., "Camera-based navigation of a low-cost quadrocopter In Intelligent Robots and Systems (IROS)", 2012 IEEE/RSJ International Conference, October 2012, pp. 2815-2821.

[11]Hrishikeshavan V, Chopra I. Refined lightweight inertial navigation system for micro air vehicle applications American Helicopter Society International - 6th Ahs International Specialists' Meeting On Unmanned Rotorcraft System 2015: Platform Design, Autonomy, Operator Workload Reduction and Network Centric Operations. 200-210.

[12]ETH Zuric Autonomous Systems Lab, Sensor Fusion Framework. Retrieved: http://bit.ly/2jtVDft

[13]Gazebo: Robot simulation made easy, `gazebosim.org`, 2017.

[14]Intrinsyc: Qualcomm Snapdragon Flight, `https://www.intrinsys.com/vertical-development-platforms/qualcomm-snapdragon-flight/`, 2017.

[15]Madyastha, V.K., Ravindra, V.C., Mallikarjunan, S., and Gopalratnam, G., "Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation," in Proceedings of the AIAA Guidance, Navigation and Control Conference, Portland, Oregon, USA, Aug. 2011.

[16]R. Mur-Artal, J. M. M. Montiel and J. D. Tards, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015.

[17]Phillips, B., Hrishikeshavan, V., Rand, O., and Chopra, I., Design and Development of a Scaled Quadorotor Biplane with Variable Pitch Proprotors for Rapid Paylaod Delivery, American Helicopter Society 72nd Annual Forum,West Palm Beach, FL, May 2016.

[18]PX4 Autopilot: The professional autopilot, `dev.px4.io+`, 2017

[19]PX4 discussion forum, discuss.px4.io, 2017.

[20]Qualcomm Developer Foru", `https://developer.qualcomm.com/forum`, 2017.

[21]Qualcomm Hexagon, `https://developer.qualcomm.com/software/hexagon-dsp-sdk/dsp-processor`, 2017.

[22]Hexagon SDK 3.0, `Made available by Qualcomm with download of Hexagon 3.0 SDK`, 2016.

[23]Qualcomm Snapdragon, `https://en.wikipedia.org/wiki/Qualcomm_Snapdragon`, 2017.

[24]QR-Linux Overview, `https://wiki.codeaurora.org/xwiki/bin/QR+Linux/WebHome`, 2016.

[25]Robot Operating System, `wiki.ros.org`, 2017.

[26]ROS Answers, `answers.ros.org`, 2017.

[27]Trawny, N. and Roumeliotis, S.I., "Indirect Kalman filter for 3D attitude estimation, " University of Minnesota, Dept. of Comp. Sci. Eng., Technical Report.

[28]Intrinsyc IoT, Qualcomm, Snapdragon Flight Kit (Developer's Edition) Quick Start Guide, 2014.

[29]Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. 2002. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In Eighteenth national conference on Artificial intelligence, Rina

Dechter, Michael Kearns, and Rich Sutton (Eds.). American Association for Artificial Intelligence, Menlo Park, CA, USA, 593-598.

[30]Snapdragon 801 SoC Image, http://i-cdn.phonearena.com/images/articles/114441-thumb/snapdragon-801-soc-image-update.png, 2017.

[31]Qualcomm Snapdragon Flight Developer Guide, October, 2016.

[32]Snapdragon Flight - Board Pin-outs, 2015.

[33]Qualcomm Snapdragon Flight Reference Platform User Guide, January, 2016.

[34]Snapdragon Flight Board Specifications, https://developer.qualcomm.com/hardware/snapdragon-flight/board-specs, January, 2017.

[35]Qualcomm Snapdragon Flight User Guide, October, 2016.

[36]Sola, J., "Quaternion Kinematics for Error State Kalman Filter,", 2012. Retrieved: www.iri.upc.edu/people/jsola/JoanSola/objectes/notes/kinematics.pdf

[37]Windolf, M., Gotzen, N., Morlock, M., Systematic accuracy and precision analysis of video motion capturing systems – exemplified on the Vicon-460 systems.