PIXHAWK: A System for Autonomous Flight using Onboard Computer Vision

Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer and Marc Pollefeys

Abstract— **We provide a novel hardware and software system for micro air vehicles (MAV) that allows high-speed, low-latency onboard image processing. It uses up to four cameras in parallel on a miniature rotary wing platform. The MAV navigates based on onboard processed computer vision in GPS-denied in- and outdoor environments. It can process in parallel images and inertial measurement information from multiple cameras for multiple purposes (localization, pattern recognition, obstacle avoidance) by distributing the images on a central, low-latency image hub. Furthermore the system can utilize low-bandwith radio links for communication and is designed and optimized to scale to swarm use. Experimental results show successful flight with a range of onboard computer vision algorithms, including localization, obstacle avoidance and pattern recognition.**

I. INTRODUCTION

We present a novel small rotary wing hardware and software system design capable of autonomous flight using onboard processing for computer vision. Our system (Fig. 1) is a micro air vehicle that can be operated in- and outdoors in GPS-denied environments. Our key contribution is a lightweight system design pattern which better suits micro air vehicle applications than larger robotic toolkits geared towards ground robotics. The presented software and hardware system is an open-source research platform which enables full onboard processing on a micro air vehicle. In contrast to previous research, it allows the vehicle to be navigated fully autonomously without any radio link or external processing device. The system design allows to utilize up to four cameras (for example as two stereo camera pairs) for localization, pattern recognition and obstacle avoidance. Cameras and inertial measurement unit (IMU) are hardware synchronized and thus allow tight vision-IMU fusion. We show the validity of the system design with real flight results in the final section.

A. Onboard Processing

Current micro air vehicle research systems are using either GPS/Inertial Navigation System (INS) navigation on a microcontroller or computer vision and laser ranging using off-board processing for localization and maneuvering. Off-board processing effectively makes the MAV dependent on the external processing unit and severely limits the safety and operation range of the vehicle. In addition, the physical wireless bandwidth limits the number of vehicles which can operate in parallel. Therefore the swarm size is physically limited to a very few vehicles. This fundamental limitation can be addressed by onboard computer vision. Up until now only larger systems (20 - 100 kg all-up weight) have processed images for vision-based localization onboard. Our system brings the multi-process architecture and onboard processing capabilities from the 20-100 kg range to vehicles around 1 kg liftoff-weight. In contrast to systems using local stabilization approaches on specialized microcontroller hardware (Parrot ARDrone), the system is geared towards global localization and autonomous exploration of unknown environments using stereo vision. The presented initial results show that our system consumes at 30 Hz frame rate only 10 % of the maximum CPU load (5 ms processing time per frame) for autonomous marker based flight and 40-60 % load (20 ms processing time per frame) for stereo-based obstacle avoidance and pattern recognition, which leaves enough capacity for future work, including simultaneous localization and mapping.

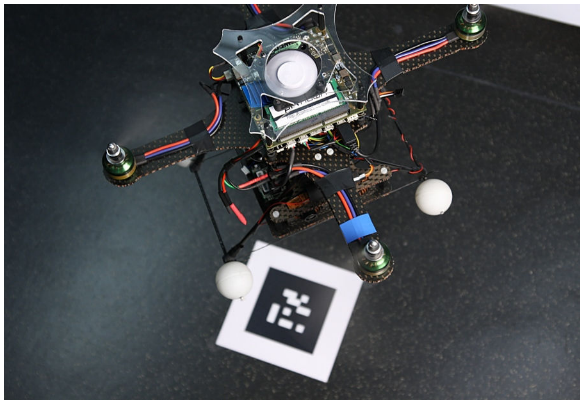


Fig. 1. PIXHAWK Cheetah Quadrotor

B. Time Base for Computer Vision

GPS and, to a large extend, laser based systems can offer a deterministic processing time to fuse the sensor data into a localization. Computer Vision in contrast has varying and in comparison often longer processing time depending on the image content. Therefore the estimation and control steps cannot depend on a fixed interval length Δt and a fixed processing delay Δp. Instead they must use the actual timestamp of all measurements to calculate the correct intervals. Therefore all information in our system is timestamped with microseconds resolution. The system guarantees correct time information (*stamp and predict design pattern*)

Lorenz Meier is master student with the Computer Vision and Geometry Lab, ETH Zurich, 8092 Zurich, Switzerland lm@student.ethz.ch

P. Tanskanen, F. Fraundorfer and M. Pollefeys are with the Computer Vision and Geometry Lab, ETH Zurich, 8092 Zurich, Switzerland {petri.tanskanen, friedrich.fraundorfer, marc.pollefeys}@inf.ethz.ch

instead of guaranteeing a fixed interval. It also relaxes the timing constraints for the computer vision algorithms; thus allowing more complex approaches which can deal with larger environments.

C. Related Work

Current MAV research is using either GPS/INS navigation on a microcontroller ([7], [6]) or computer vision/laser ranging and offboard processing. Until now, only larger systems processed images onboard. Conte [3] et al. processed visual odometry at 4 Hz on an Yamaha RMAX helicopter UAV with over 20 kg payload (94 kg maximum total weight) and 3.6 m diameter, but did not use the output for flight control. In the MAV domain previous onboard processing approaches ([4]) used simple optical flow mouse sensors to locally stabilize the position and to do simple obstacle avoidance. Other approaches used laser scanners and cameras but did not process the data/images onboard. The system of [1] uses an Asctec Pelican Quadrotor with Hokuyo URG line scanner and onboard Intel ATOM processor to collect laser scan data and camera images for off-board processing. This data is sent to an off-board notebook for processing the actual localization. The outdoor system of [7] uses an analog TV camera for object tracking and GPS/INS for position control. The work of [2] et al. demonstrated visual localization using a camera on a USB tether cable and processing on a notebook. The STARMAC quadrotor [6] has a PC104 form factor onboard computer, but does not utilize it for vision processing due to limited performance.

II. Computer Vision Framework

Following the well-known principle of stereo cameras, to estimate the metric distance in 3D, two images with a known baseline (distance of the camera centers) are needed. Therefore the PIXHAWK quadrotor has a setup of 2x2 cameras in two stereo setups, pointed down and front with 5 cm baseline. All four cameras are triggered from the onboard inertial measurement unit. Computer vision allows to extract both the 3D geometry of the environment as well as its texture/appearance. Therefore multiple algorithms are necessary to extract all information of an image, leading to the need to distribute the images to multiple algorithms in multiple processes. Previous approaches ([1], [2]) did not have the possibility to run camera interface and image processing pipelines separated. The central image hub distributes the combined images and IMU information to all connected computer vision algorithms, including localization and pattern recognition algorithms.

A. Vision-IMU Combination

As the trigger system supports both monocular and stereo setups, any localization approach that uses one or two cameras can receive the vision-IMU data and process it. The attitude (roll, pitch, yawspeed) is available as part of the image metadata. This allows to speed up the localization algorithms, either by providing an initial guess of the attitude or in closed form as direct contribution to the localization. In case of the 3-point algorithm [5], the calculation steps for the triangulation are significantly simplified when substituting parts of the calculation by IMU roll and pitch, which speeds up RANSAC (see Fig. 2 for the geometric relation). For any non-global vision based localization approach IMU information can provide the gravity vector and heading as global reference. This property is important when local vision information is used as controller input.

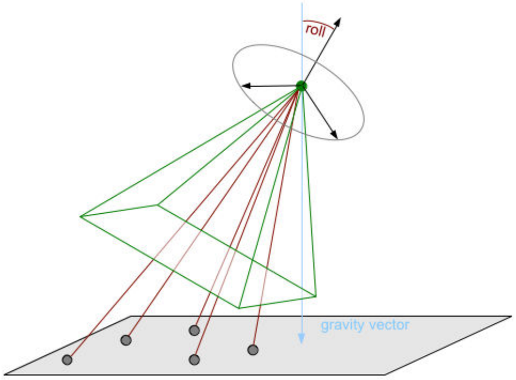


Fig. 2. Relation of gravity vector and camera angles

III. Aerial Robotics Middleware

Several toolkits for larger unmanned ground, surface and air vehicles have been widely used in research. Micro air vehicles with significant or complete onboard processing are a rather new development, though. Existing toolkits for ground robotics include ROS, CARMEN and CLARAty. The communication architecture significantly blocks ground robotic toolkits to be adapted on small-scale flying platforms. All toolkits assume TCP/IP or UDP network links, such as IEEE 802.3 Ethernet and IEEE 802.11a/b/g/n WiFi. However, MAV onboard-networks typically include several devices connected via serial links. As these toolkits do not scale down to this link type, every packet has to be transcoded by bridge processes, leading to unnecessary effort and system complexity. Therefore we propose a new communication protocol and architecture that can be transparently used on different hardware links and minimizes the system complexity.

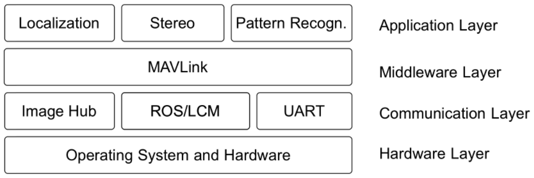


Fig. 3. PIXHAWK Middleware Architecture Layers

The PIXHAWK robotics toolkit is based on a lightweight protocol called MAVLink, which scales from serial to UDP links. It serves also as communication protocol between flight computer (pxIMU) and onboard main computer (pxCOMex/pxOvero). This is also important for the safe operation of any robotic aircraft, as a human operator should always be able to intervene. The typical 30-100 m range of WiFi does not generalize to most outdoor applications, which makes a communication architecture scaling down to radio modems also desirable for the off-board communication. As shown in Figure 3, the PIXHAWK Linux middleware consists of several layers. This architecture allows to use the different base communication layers (ROS and LCM) and provides a convenient high-level programming interface (API) to the image distribution server. MAVLink messages from the IMU and the ground control station can also be directly received in any process.

A. *MAVLink Protocol*

Our MAVLink protocol is a very lightweight message marshalling protocol optimized for micro air vehicles. It has only 8 bytes overhead per packet, allows routing on an inter-system or intra-system level and has an inbuilt packet-drop detection. Due to the low overhead, it is both suitable for UDP and UART/radio modem transport layers. The efficient encoding also allows to execute the protocol on microcontrollers. These properties allowed building a homogenous communication architecture across the PIXHAWK system. MAVLink has been already adopted by a number of other systems (pxIMU autopilot, ArduPilotMega autopilot, SLUGS autopilot, UDB autopilot). The MAVLink sentences are generated based on an XML protocol specification file in the MAVLink format. The code generator ensures well-formed messages and generates C89-compatible C-code for the message packing and unpacking. This allows fast and safe extensions and changes to the communication protocol and ensures that no implementation errors will occur for new messages. Our current implementation supports the use of the lightweight communication marshalling library (LCM) or the Robot Operating System (ROS) as transport layers.

IV. Vehicle Design

The PIXHAWK Cheetah quadrotor design was built from scratch for onboard computer vision. Beside the commercial-off-the-shelf (COTS) motor controllers and cameras all electronics and the mechanical frame is our custom design. First the payload, consisting of the pxCOMEx processing module and four PointGrey Firefly MV USB 2.0 cameras, was selected. The system design then followed the requirements of onboard computer vision.

A. *Electronics*

The onboard electronics consists of an inertial measurement unit and autopilot unit, pxIMU, and the onboard computer vision processing unit, pxCOMEx.

1) Autopilot Unit: The pxIMU inertial measurement unit/autopilot board (Fig. 4) provides 3D linear acceleration (accelerometer, ±6g), 3D angular velocity (±500 deg/s), 3D magnetic field (± milligauss), barometric pressure (130-1030 Hectopascal (hPa)) and temperature. The onboard MCU for sensor readout and sensor fusion as well as position and attitude control is a 60 MHz ARM7 microcontroller. It can be flashed via an USB bootloader and stores settings such as PID parameters in its onboard EEPROM. It provides the required I2C bus to the motor controllers and additional GPIOs, ADC input and other peripherals. It is interfaced via UART to the computer vision processing unit and it operates at a maximum update rate of 200-500 Hz.

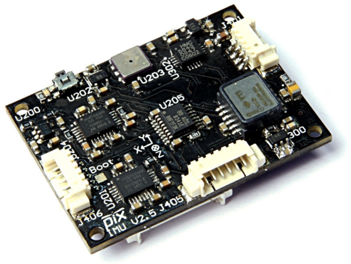


Fig. 4. pxIMU Inertial Measurement Unit

2) Processing Unit: The processing unit it the core piece of the system and consists of a two-board stack. The pxCOMEx base board provides the USB and UART peripherals to interface machine vision cameras, communication equipment and the pxIMU module. It can accept any micro COM express industry standard module. Currently, a Kontron etxExpress module with Intel Core 2 DUO 1.86GHz and 2 GB DDR3 RAM is used, but future upgrade options include Intel i7 CPUs. It has 4x UART, 7x USB 2.0 and 1x S-ATA 2.0 peripheral options. The typical onboard setup consists of 4x PointGrey Firefly MV monochrome, 1x USB 2.0 802.11n WiFi adapter and 1x S-ATA 128 GB SSD with more than 100 MB/s write speed. The pxIMU unit, the GPS module and the XBee radio modem are connected via UART to the processing unit. With a weight of 230 g including cooling and only 27 W peak power consumption, the processing unit can be easily lifted by a wide range of aerial systems, not limited to the quadrotor presented here.

B. *Mechanical Structure and Flight Time*

Our custom mechanical design effectively protects the onboard processing module in case of a system crash and the fixed mounting of the four cameras allows inter-camera and camera-IMU calibration. As the processing board and four cameras represent a relatively large payload of 400 g for the small diameter of 0.55 m (0.70 m for the larger version) of the quadrotor, the overall system structure has been optimized for low weight. It consists of lightweight carbon sandwich material with carbon fiber base plates and an inner layer made of Kevlar in shape of honeycombs. Each of the four motors with 8” propeller contributes a maximum of 452g thrust, enabling the system to lift 400g payload at a total system weight of 1.00–1.20 kg, including battery. This allows a continous flight time of 7-9 minutes with 8” propellers and 14-16 minutes with 12” propellers. The propulsion consumes 150-180W for hovering, while the highspeed onboard computer consumes only 27 W peak. Therefore flight time is governed by the weight of the system.

V. Localization and Flight Control Pipeline

The localization and flight control pipeline is only one of the several onboard pipelines. As the PIXHAWK middleware provides a precise time base, a standard textbook estimation and control pipeline already performs well for autonomous flight. The overall pipeline, including camera interfacing and communication, consumes only 10–15 % of the total CPU power. Other implemented pipelines are stereo obstacle avoidance and planar pattern recognition. Individual pipelines can be activated / deactivated at runtime and individual pipeline components can be replaced by different algorithms without changes to the overall system. Fig. 5 illustrates the data processing and information flow from image capture to motor control output.

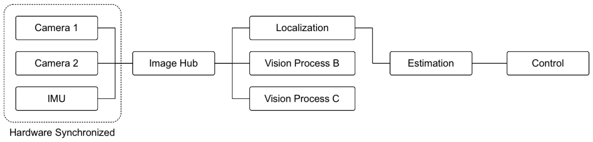


Fig. 5. Vision-guided Control Pipeline

A. Vision-IMU Synchronization

As the vision-IMU fusion depends on measurements from a known timebase, the image capture is controlled by the inertial measurement unit using a shutter signal. The IMU also delivers the current roll, pitch and yaw estimate at the time of image capture with the shutter time to the vision pipeline. Images are transmitted over USB to the camera process, while the IMU measurements and the shutter timestamp are delivered through a serial interface. Image transmission from camera to main memory via USB takes approximately 16 ms, the transmission of the shutter timestamp from IMU to main memory via serial/MAVLink takes approximately 1–3 ms. As it is guaranteed that the IMU data arrives earlier than the image, the camera process can always deliver the full vision-IMU dataset at once to the localization process.

B. Vision based Localization

The initial flight tests were conducted using a marker based approach with an adapted implementation of ARToolkit+ [8] for the localization. The marker positions are encoded in a global world map with the 6D position and attitude of each marker. By extracting the marker quadrangle, the global marker position can be estimated by calculating the homography on the four corner points of the quadrangle. The correct orientation on the quadrangle plane and the marker ID is encoded by a 2D binary code inside each marker. An example of a larger marker setup is shown in Fig. 6. However, the system itself is not depended on this particular approach – any kind of localization algorithm can be used.

C. Outlier Removal

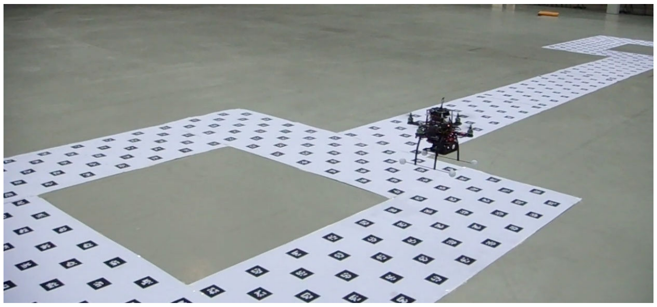


Fig. 6. Figure-8 setup at IMAV 2010 competition

The data is filtered with a Kalman filter in the next step, which implies that the filter is parametrized with the error model of the computer vision approach. As IMU and vision both estimate the 3 degree of freedom attitude of the helicopter, this redundant data can be used to detect and remove position outliers produced by the localization step. Any erroneous vision measurement will not only contain a wrong position estimate, but also wrong attitude estimate because of the projective dependency of position and attitude. Position outliers can therefore be rejected based on the comparison of roll and pitch estimates from the IMU and from the visual localization.

D. Discrete Kalman Filtering

The remaining data is more conformant to the normal distribution, and allows the use of a simple discrete Kalman filter. As the dynamics of a quadrotor are only loosely coupled in x, y and z direction, the dynamics can be modelled as three independent dimensions. As the yaw angle is of much better accuracy and resolution in indoor settings from computer vision than from a magnetometer (due to iron structures in the building), the yaw angle is taken as the fourth independent dimension for filtering. Given this quadrotor dynamic model, the Kalman filter is designed as a block of 4x 1D Kalman filters with position and speed as states. The Kalman filter assumes a constant speed model, and takes position as input. The estimated velocity is critical to damp the system, as the only physical damping is the air resistance on the horizontal plane, which is not significant at the typical hovering or low-speed conditions. The states of the four Kalman filters are:

$$x_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \qquad y_k = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \qquad z_k = \begin{bmatrix} z \\ \dot{z} \end{bmatrix} \qquad \psi_k = \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix}$$

We try to estimate the current state of the vehicle x_k, which is modeled by

$$x_k = A \cdot x_{k-1} + w_{k-1}.$$

Where the dynamics matrix A models the law of motion, x_{k-1} is the previous state and w_{k-1} the process noise. This motion is measured at certain time steps, where the measurements are expressed as the gain H times the current state plus the measurement noise v.

$$z_k = H \cdot x_k + v_k$$

The speed in the model will therefore only be changed by measurements and then again assumed constant during prediction. From this formulation it is already obvious that varying time steps can be handled by the filter as long as they are precisely measured. As this filter does not include the control input matrix B, the filter is assuming a constant speed model, which is a valid approximation if the filter update frequency is fast enough with respect to the change of speed of the physical object. Because the PIXHAWK system provides a precise timebase, the filter uses the measured inter-frame interval as time difference input Δt. It is using the standard Kalman prediction-update scheme. If measurements are rejected as outliers, the filter only predicts for this iteration and compensates in the next update step for the then longer time interval. This allows the system to estimate its egomotion for several seconds without new vision measurements.

E. Position and Attitude Control

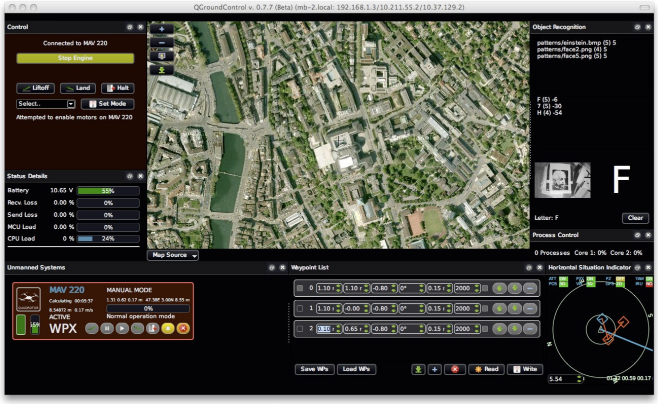
As already pointed out, the x, y, z, yaw motion can be modeled as independent. Thus, it is possible to control the quadrotor's x- and y-position with the angle of attack of the collective thrust vector by setting the desired pitch angle for x and the desired roll angle for y. The z-position can be controlled with the component of the collective thrust collinear to the gravity vector. The yaw angle finally can be controlled by the difference of rotor drag of the clockwise (CW) and counter-clockwise (CCW) rotor pairs. As the previous step contributed a smoothened position and speed estimate with little phase delay, no model-driven/optimal control is needed to account for the missing direct speed measurement. The controller can thus be designed as a standard PID controller, implemented as four independent SISO PID controllers for x, y, z, and yaw. Attitude control was implemented following the standard PID based attitude control approach for quadrotors using one PID controller for roll, pitch and yaw each. The craft is actuated by directly mixing the attitude control output onto the four motors.

VI. Operator Control Unit

The design paradigm presented in this paper shows a clear separation of the onboard autonomy system and the off-board operator control. As the MAV is not any more a remote sensor head, no sensor data needs to be transmitted to the ground control station for autonomous flight. It is therefore desirable to reduce the communication and operator load and only send abstract system information such as remaining fuel/battery and position. The QGroundControl application allows to represent multiple vehicles. The map view from Fig. 7 shows the aerial map view.

VII. Experimental Results / Flight

Fig. 8 shows flight results using artificial marker based localization. The plot shows a flight around a rectangular path. The two perpendicular movements are autonomous startoff and landing. The right vertical trajectory is the open-loop liftoff. As initially no computer-vision based localization is available, the helicopter runs an open-loop maneuver until it has 0.5 m altitude. The left vertical path is the autonomous landing after a short period of hovering on a spot. The plot shows a section of the Figure-8 setup of the International Micro Air Vehicle Competition 2010, which is also depicted in Fig. 6. A video of a similar flight can be viewed at `http://pixhawk.ethz.ch/videos/`.


QGroundControl v. 0.7.7 (Beta) (mb-2.local: 192.168.1.3/10.211.55.2/10.37.129.2)

Connected to MAV 220

Attempted to enable motors on MAV 220

Battery 10.65 V 55%
Recv. Loss 0.00 % 0%
Send Loss 0.00 % 0%
MCU Load 0.00 % 0%
CPU Load 0 % 24%

patterns/einstein.bmp (5) 5
patterns/face2.png (4) 5
patterns/face5.png (5) 5

F (5) -6
7 (5) -30
H (4) -54

Letter: F

0 Processes Core 1: 0% Core 2: 0%

MAV 220 MANUAL MODE
Calculating 00:05:37 1.31 0.62 0.17 m 47.38E 3.00N 8.55 m
8.54872 m 0.17 m/s 0%
ACTIVE Normal operation mode
WPX

0 1.10 1.10 -0.80 0° 0.15 2000
1 1.10 -0.00 -0.80 0° 0.15 2000
2 0.10 0.65 -0.80 0° 0.15 2000

5.54 01:32 00.59 00.17

Fig. 7. QGroundControl Operator View with map and detected patterns

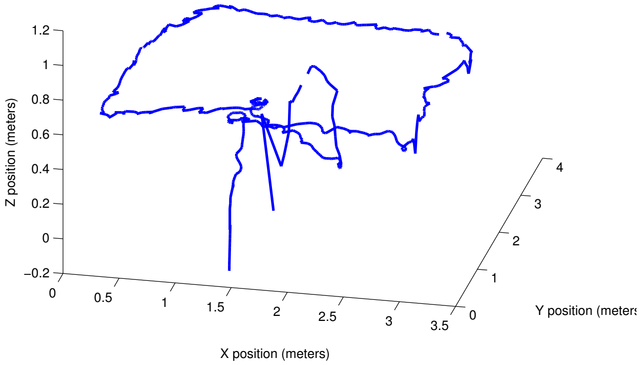


Fig. 8. Trajectory of an autonomous flight using ARToolkit localization including takeoff and landing

As our system is modular, we can also easily replace the ARToolkit-based localization with other methods for position control. Fig. 9 shows a trajectory resulting from a flight using a Vicon motion capture system. The localization is very precise ($< 1mm$ error) at a maximum rate of 250 Hz. We use a low latency wireless link to transfer the current position computed by the motion capture system to the helicopter. The flight trajectory contains autonomous takeoff and landing, as well as several yaw rotations and an obstacle avoidance reaction (overlapping parts of the trajectory on top of the figure). The onboard view of the helicopter during this flight is shown in Fig. 10. This illustrates the flexibility and scalability of the system, as these experimental results include onboard stereo depth estimation, dynamic obstacle avoidance and pattern recognition operating in parallel on the onboard computer. Although sensors delivering a depth map without high processing burden currently emerge, such as the PrimeSense Kinect sensor, for outdoor applications stereo still remains the best option to obtain a depth map on a micro air vehicle. The significant higher processing performance of the presented platform is therefore a key differentiator to the previous state of the art.

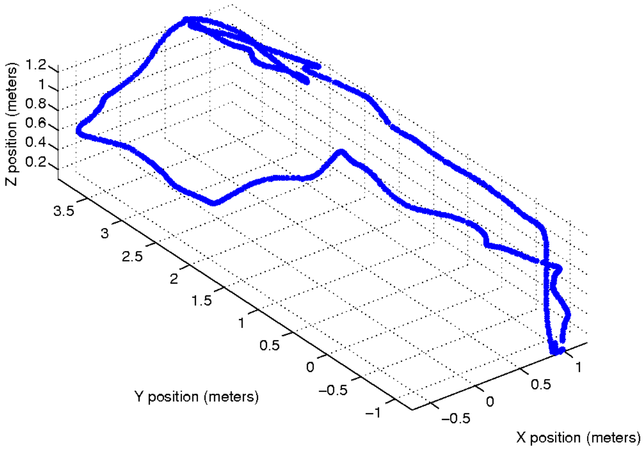


Fig. 9. Trajectory of an autonomous flight using Vicon localization including takeoff and landing

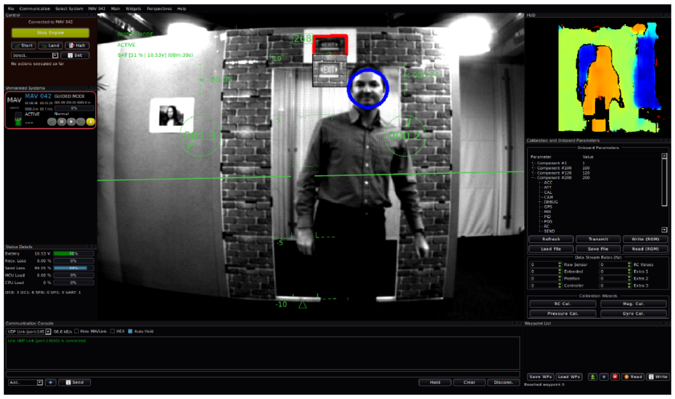


Fig. 10. QGroundControl HUD View with live image streaming from the helicopter. The live view shows detected faces and patterns by the helicopter. On the right top corner the computed depth map from the stereo camera setup is shown.

VIII. CONCLUSIONS AND FUTURE WORKS

The open-source system presented in this paper provides a new onboard and offboard architecture for computer vision based flight with tight IMU integration. As the PIXHAWK system is built for onboard computer vision, future work will focus on natural feature based localization and mapping for autonomous flight. Other future improvements are the optimization of the onboard position and attitude control and the extension of the current waypoint based scheme to trajectory control. The current system load for artificial feature based localization is 10 % of the maximum CPU capacity. Higher-level approaches such as stereo obstacle avoidance and pattern recognition however increase significantly higher load in in the 40-60 % range if run in parallel. Because of the choice of an industry standard computing platform, the presented system will scale with future increases in processing performance and always roughly deliver the processing speed of a medium-level notebook computer. Hence enough capacity for extensions is available onboard. As the system design provides a precise common time base, IMU-GPS-vision fusion will be a future extension for outdoor navigation. On a multi-system level, the lightweight MAVLink protocol provides an ideal basis for future swarm extensions. As all processing is onboard, the number of vehicles is not limited by the communication bandwidth.

IX. ACKNOWLEDGMENTS

We would like to thank our students (in alphabetical order) Bastian Bücheler, Andi Cortinovis, Fabian Landau, Laurens Mackay, Tobias Nägeli, Philippe Petit, Martin Rutschmann, Amirehsan Sarabadani, Christian Schluchter and Oliver Scheuss for their contributions to the current system and the students of the previous semesters for the foundations they provided. Raffaello d'Andrea and Sergei Lupashin (ETH IDSC) provided constant and valuable feedback. This work was supported in part by the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant #231855 (sFly) and by the Swiss National Science Foundation (SNF) under grant #200021-125017.

REFERENCES

[1] M Achtelik, A Bachrach, R He, and S Prentice. Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. *Proceedings of the SPIE Unmanned Systems Technology XI, 7332, 733219*, Jan 2009.

[2] M Blösch, S Weiss, D Scaramuzza, and R Siegwart. Vision based mav navigation in unknown and unstructured environments. *Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[3] G Conte and P Doherty. An integrated uav navigation system based on aerial image matching. *Proceedings of the IEEE Aerospace Conference*, Jan 2008.

[4] SG Fowers, Dah-Jye Lee, BJ Tippetts, KD Lillywhite, AW Dennis, and JK Archibald. Vision aided stabilization and the development of a quad-rotor micro uav. *International Symposium on Computational Intelligence in Robotics and Automation, 2007. CIRA 2007.*, pages 143 –148, 2007.

[5] F Fraundorfer, P Tanskanen, and M Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6314, pages 269–282. Springer Berlin / Heidelberg, 2010.

[6] G Hofiann, D Rajnarqan, and S Waslander. The stanford testbed of autonomous rotorcraft for multi agent control (starmac). *Proceedings of Digital Avionics Systems Conference (DASC04).*

[7] N Roy, R He, A Bachrach, and M Achtelik. On the design and use of a micro air vehicle to track and avoid adversaries. *International Journal of Robotics Research*, Jan 2010.

[8] D Wagner and D Schmalstieg. Artoolkitplus for pose tracking on mobile devices. *Proceedings of 12th Computer Vision Winter Workshop*, Jan 2007.