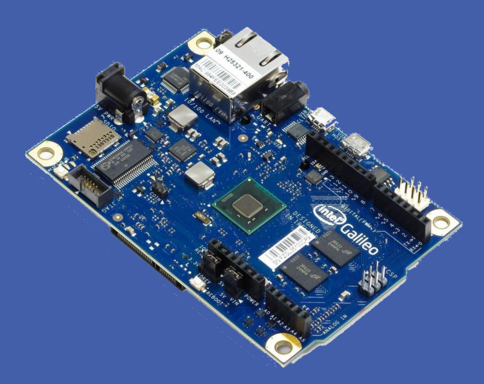


Intel's Galileo IoT board

Fleye on the Car: Big Data meets the Internet Of Things

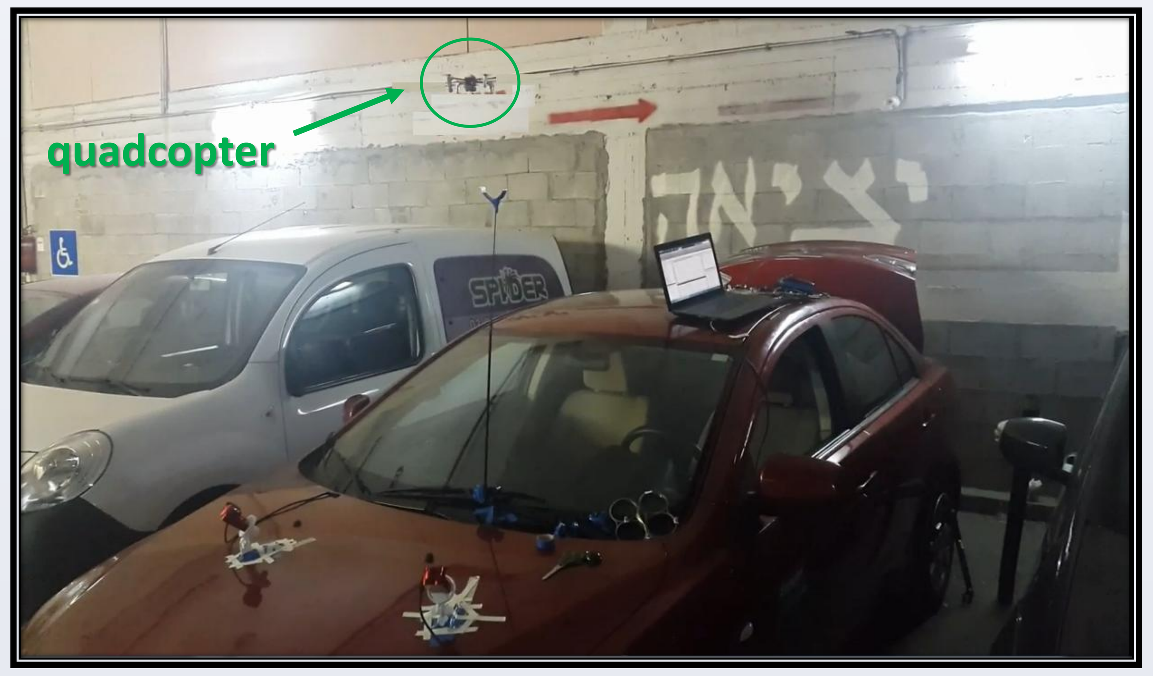
Soliman Nasser [1] Andrew Barry [2] Daniela Rus [2] Guy Peled [1]

Guy Rosman [2] Marek Doniec [2] Mikhail Volkov [2] Dan Feldman [1]

The Walkera's Ladybird quadcopter

Fleye

An open-code system that consists of an autonomous drone (nano quadrotor) that carries a radio camera and flies few meters on top and outside the car.



- Real-time streaming video from the view of a hovering quad-copter above the car
- Video and other sensors data processed on the cloud
- The output video is then projected to the smart glasses of the driver
- The driver control the quad-copter using voice commands
- Based on low-cost Internet of Things (IoT) hardware

The **Fleye** system during a live video stream from the autonomous quadcopter to the smart glasses of the driver

SYSTEM OVERVIEW

The hardware of our system consists of the following commercial products:

- **Intel's Galileo**: A wearable mini-computer ("Internet of Thing" board), similar to Raspberry pie.
- **Walkera's Ladybird**: a small toy drone (quadcopter). No sensors or data transmitters, only a receiver.
- Transmitter: **Walkera DEVO 7E Transmitter** for sending commands to the quadcopter (Comes in the same package of the quadcopter).

- **Arduino UNO R3**: cheap that sends PPM signals to the remote control of the quadcopter and controls it.

- **TX5805**: analog radio low-weight FPV (pilot view) camera that is mounted on the quadcopter and sends radio 5.8GHz video signal.

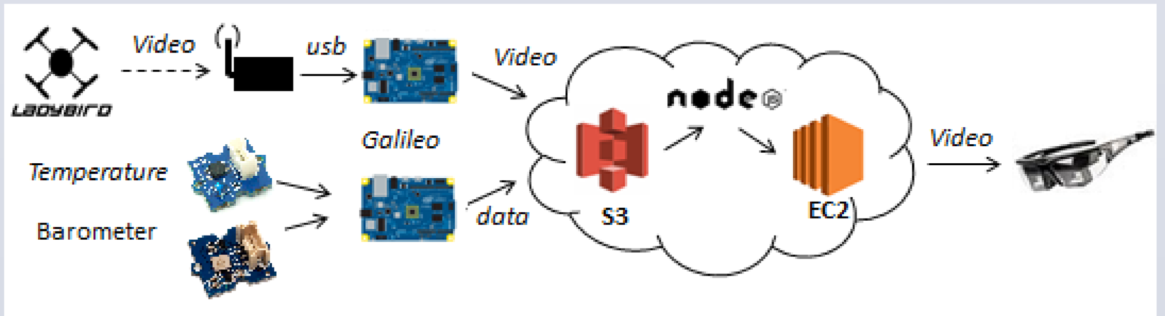
- **RC5805**: 5.8Ghz Video receiver that gets the analog streaming video from the TX5805.

- **Diamond VC500**: video grabber converts the analog composite video from the RC5805 to a digital video through a USB.

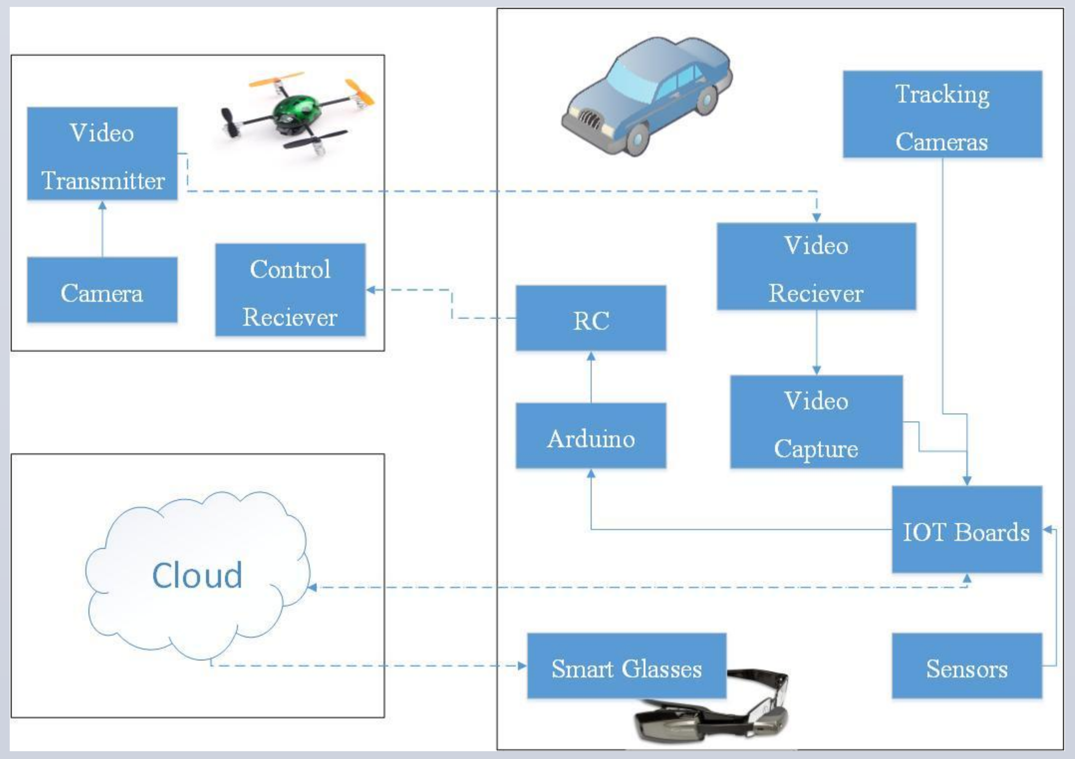
- **Cameras** (Optitrack Flex 3 or Logitech C920 Web-cam): for tracking the quadcopter.

- **VUZIX m100**: smart glasses for the driver.

- **Amazon's web services**: for running computer vision algorithms on the EC2 cloud.



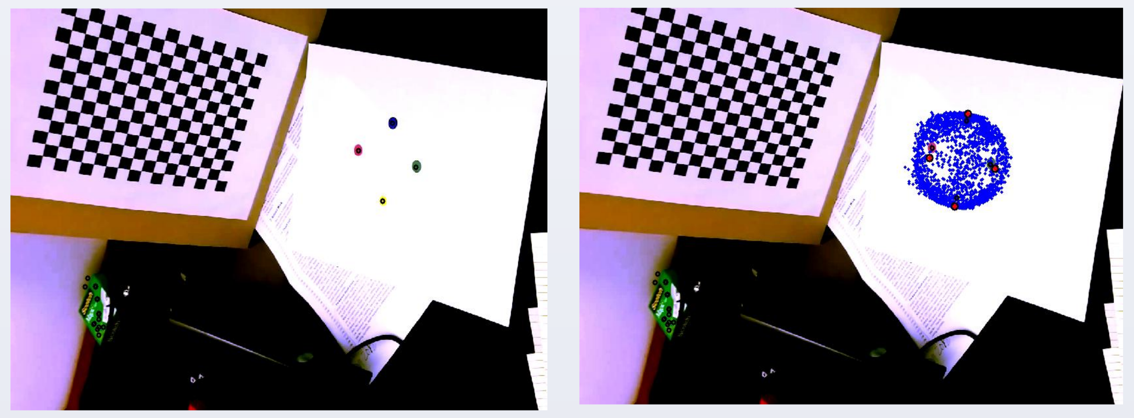
The hardware setup of Fleye.



Data flow of the Fleye system

LOCALIZATION AND TRACKING

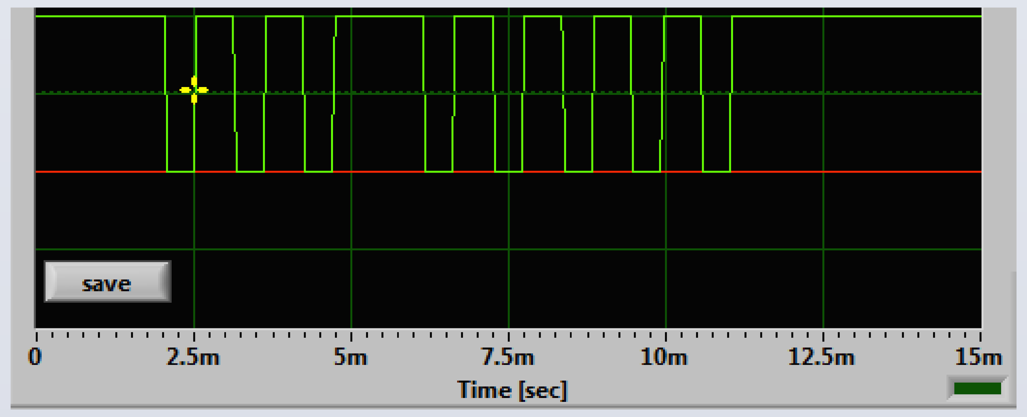
- A novel tracking and localization algorithm that identifies the position and orientation of our quadcopter
- 6 degrees of freedom (x, y, z, pitch, yaw, roll)
- The quad-copter is equipped with known colored markers



Tests on the tracking module

AUTONOMOUS QUADCOPTER

- Pulse position modulation (PPM) signal
- Interrupt-driven PPM controller using a Arduino Uno
- The input PPM signal is a 33:3 kHz 8-channel
- Pulse width of 400 micro-seconds and channel values between 600 and 1500 micro-seconds



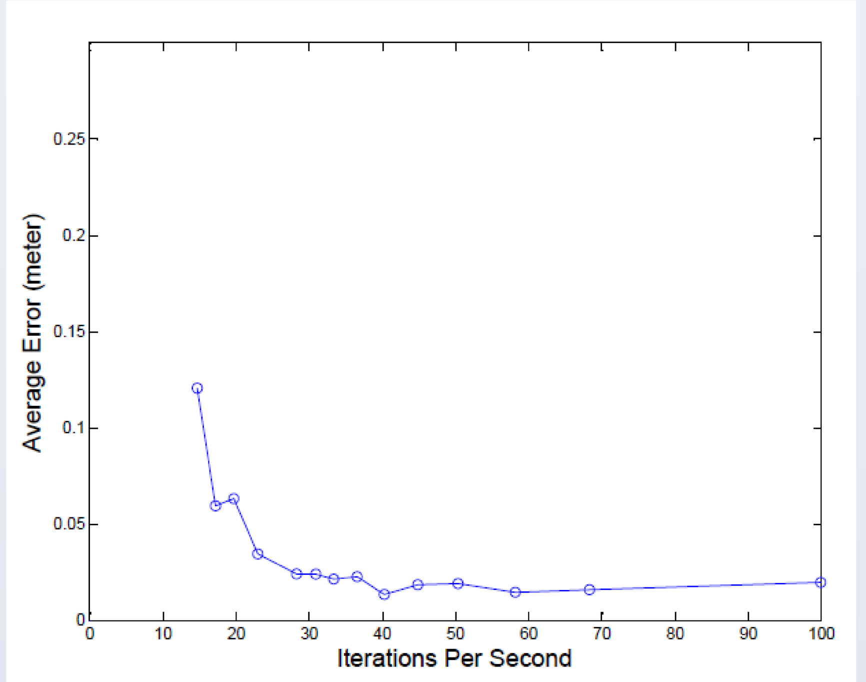
A PPM signal that is transmitted from the Arduino to the remote control

CLOUD PROCESSING

- The streaming video data from the camera on the hovering quadcopter, the cameras on the car, the glasses, and the sensors on the Galileo board are transmitted to Amazon's EC2 cloud for high performance computation.
- The result is a processed video image, possibly with additional markers and text, that is uploaded in real time to an http address.
- The glasses project this http content to the driver.
- Due to various reasons, the video captured from the quadcopter's camera is unstable and shaky. Therefore, we un video stabilization code on the cloud and then present the output video stream to the driver for a better experience.

EXPERIMENTS

In the following figure we show the summary of experiments that we did for testing the stability of the quadcopter with respect to a given computer vision algorithm, by hovering it over a given point on the ground. The timing includes the radio transmission time, the tracking time, the video capture and the PID controller. The y-axis shows the average error in meters from the quadcopter to the target position that it supposes to hover in. The x-axis shows the number of updates to the quadcopter position in the control loop of tracking and updating the current position.



Stability tests on a hovering ladybird quadcopter

REFERENCES

[1] http://arduino.cc/en/main/arduinoboarduno

[2] https://gigaom.com/2013/08/08/why-google-glass-costs-1500-now-and-will-likely-be-around-299-later/

[3] http://www.amazon.com/easycap-audio-video-capture-adapter/dp/b0019sssmy.

[4] http://www.intel.com/content/www/us/en/do-it-yourself/galileo-maker-quark-board.html.

[5] http://www.marketwatch.com/story/mobileye-shares-rally-as-rms-initiate-positive-coverage-2014-08-26.

[6] http://www.vuzix.com/consumer/products m100/.

[7] http://www.walkera.com/en/showgoods.php?id=1652.

[8] http://www.walkera.com/en/showgoods.php?id=450.

[9] S. Arya, D. M. Mount, N. S. Netanyahu,R. Silverman, and A. Y. Wu. An optimal algorithmfor approximate nearest neighbor searching fixed dimensions. J. ACM, 45(6):891-923, Nov. 1998.

[10] G. Bradski and A. Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[11] E. Dagan, O. Mano, G. P. Stein, and A. Shashua. Forward collision warning with a single camera. In Intelligent Vehicles Symposium, 2004 IEEE, pages 37-42. IEEE, 2004.

[12] K.-F. Lee. Automatic Speech Recognition: The Development of the Sphinx Recognition System, volume 62. Springer, 1989.

[13] B. Ristic, S. Arulampalam, and N. Gordon. Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, 2004.

[14] G. Stein. System and method for detecting obstacles to vehicle motion and determining time to contact therewith using sequences of images, Sept. 26 2006. US Patent 7,113,867.

[15] J. Ueki, J. Mori, Y. Nakamura, Y. Horii, and H. Okada. Development of vehicular-collision avoidance support system by inter-vehicle communications - vcass. In Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th,
volume 5, pages 2940{2945 Vol.5, May 2004.

1
Robotics and Big Data Lab, Computer Science Department, University of Haifa, Israel

2
Computer Science and Artificial Intelligence Laboratory, MIT Cambridge, MA USA