2017

Collision Avoidance for Quadcopters

Andrew Ward
Virginia Commonwealth University

Brad Clifford
Virginia Commonwealth University

Shane Gifford
Virginia Commonwealth University

Thomas White
Virginia Commonwealth University

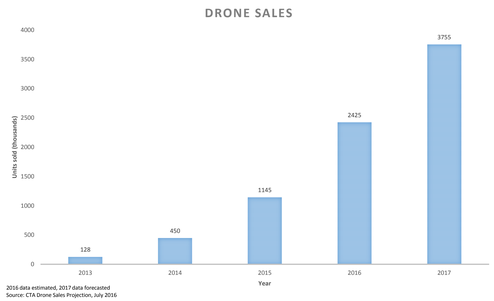
Collision Avoidance for Quadcopters

ECE 405 | Andrew Ward, Brad Clifford, Shane Gifford, Thomas White | **Faculty adviser:** Wei Zhang, Ph.D. / Tim Bakker, Ph.D.

Objective

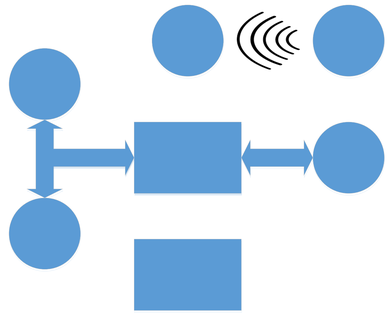
The purpose of our project is to develop a system to be mounted on a quadcopter drone which can detect potential collisions with the outside environment and navigate the quadcopter to avoid such collisions.

Example of drone obstacle avoidance



- Multirotor drones are expanding market
- Drones are expensive and fragile
- Market exists for collision avoidance system which can be mounted to different drones

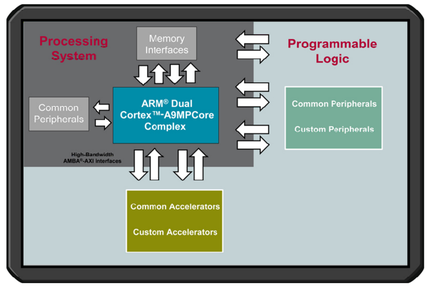
Design



Functional Block Diagram

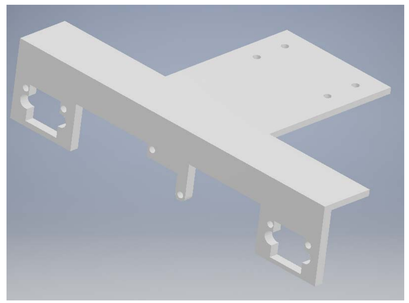
- Two cameras capturing QVGA video
- Ultrasonic sensor as failsafe for cameras
- DT7 remote control
- DR16 receiver capturing user input as SBUS signal
- Zybo Zynq-7000 development board
  - Processes sensor data with NEON integration to determine if collision is imminent
  - Overrides user control data with board-generated signal to prevent collision if necessary
  - Uses SBUS, RS232, and SCCB (I2C) bus protocols
- Naza M-v2 flight controller responsible for interfacing between control signals and motors

The microcontroller used is the Zybo Zynq-7000 System-on-a-Chip (SoC). The integrated hardware and software design approach of this board allows for easier interfacing between components, as bus interfaces can easily be added and configured in Vivado to connect our peripheral components to our board. This board also includes a NEON accelerator which allows for faster vectorized calculations.



Zynq HW/SW Co-design Diagram

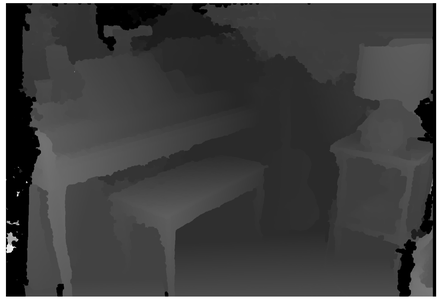
To ensure consistent sensor data and to keep our system securely in place, a mounting bracket was designed in Autodesk Inventor which holds the sensors and the Zybo in place on the quadcopter.



Prototype of mounting bracket

Conclusions

- System capable of avoiding collisions when flying slowly forward
- Performance depends on environment, speed of drone



Disparity Map generated in MATLAB

Possible directions for performance improvement

- Configure system to automatically make use of cameras' full capabilities (white balance, exposure timing, etc.)
- Include GPU component on-board to improve speed of calculations
- VHDL model development for quicker pipeline

Possible areas for further research:

- Sensors to detect obstacles in every direction
- Compare different sensor arrays' performance
- Integrate with other systems for autonomous flight
- Best path navigation/navigation efficiency