

Indoor Micro-UAV Navigation with Minimal Sensing

Dennis Melamed¹

Advisers: Professor Volkan Isler² & Professor Derya
Aksaray³

October 1, 2018

¹Dept. of Electrical and Computer Engineering

²Dept. of Computer Science and Engineering

³Dept. of Aerospace Engineering and Mechanics

OUTLINE

OVERVIEW

BACKGROUND

METHODOLOGY

Door Detection

Independent Flight

CONCLUSION

PROPOSED PROJECT & SUCCESS CRITERIA

- ▶ Fly through door, land safely
- ▶ Random initial location
- ▶ Low-quality sensors
(Camera, IMU)
- ▶ 27g weight



Success: Fly with <20% failure rate through door
(failure=crash, battery loss, missing door)

Source: <https://www.bitcraze.io/crazyflie-2/>

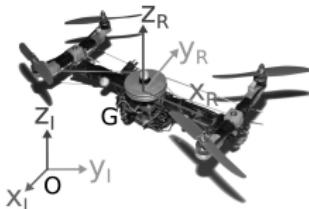
BACKGROUND

- ▶ Research with autonomous navigation in constrained spaces
 - ▶ GapFlyt
- ▶ Research with small scale UAV control in motion capture systems
 - ▶ Morphing
- ▶ Combination not well studied

Source:

Sanket et al. (2018)

Riviere et al. (2018)



(a)



(b)

PROJECT OVERVIEW

- ▶ Initial setup
 - ▶ Motion capture
- ▶ Door detection
- ▶ Independent Flight

DOOR DETECTION - HOUGH TRANSFORM

- ▶ Hough Transform for line detection
- ▶ Group close-to-parallel lines
- ▶ Choose largest rectangle formed



Source: Akgul (2015)

DOOR DETECTION - FUZZY LOGIC

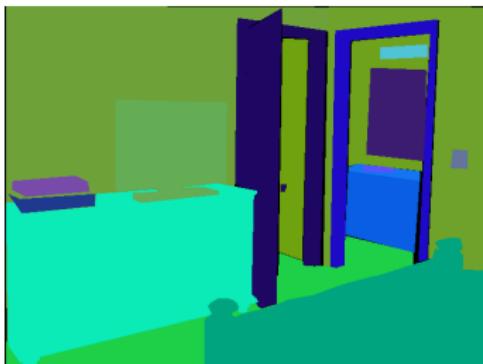
- ▶ Fuzzy concept of vertical/horizontal line
- ▶ Fuzzy concept of door frame (2 vertical + 1 horizontal)

Muñoz-Salinas et al. (2004)

DOOR DETECTION - FULLY CONVOLUTIONAL NEURAL NETWORK

- ▶ Add noise/lower resolution
- ▶ Output segmented door

$$IoU = \frac{\text{AreaOfOverlapBetweenDetections}}{\text{AreaOfUnionOfDetections}}$$



Source: Zhou et al. (2017)

INDEPENDENT FLIGHT - DEAD RECKONING

- ▶ Calculation of unit vector pointing at door
- ▶ Launch and fly quickly to minimize drift

INDEPENDENT FLIGHT - RECURRENT NEURAL NETWORK

- ▶ Train network w/ waypoint navigation ground truth
- ▶ Has memory - deal with velocities

INDEPENDENT FLIGHT - REINFORCEMENT LEARNING

- ▶ Similar inputs/outputs as regular RNN
- ▶ Train without ground truth, reward instead:
 - ▶ D - distance from door
 - ▶ R - is UAV orientation acceptable
 - ▶ O - distance to obstacles

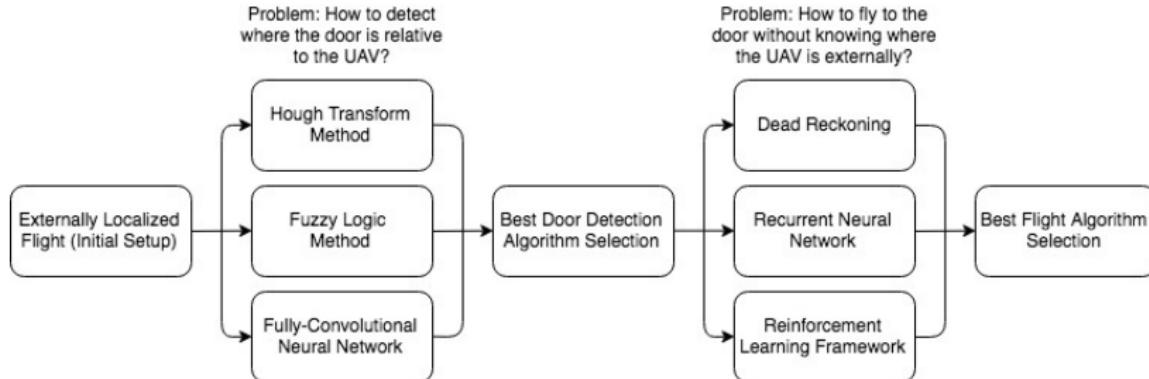
$$Reward = K_1/D + K_2 * R + K_3 * O + K_n * OtherFactors$$

BUDGET & SCHEDULE

- ▶ Small budget for wood to build door (\$20)
- ▶ Oct 25 - Initial setup and UAV flight
- ▶ Dec 12 - Door detection algorithms compared
- ▶ Apr 05 - Independent flight algorithms compared

CONCLUSION

- ▶ Explore intersection of small size, auto navigation
- ▶ Compare methods for door detection
- ▶ Compare methods for independent flight



Questions

REFERENCES

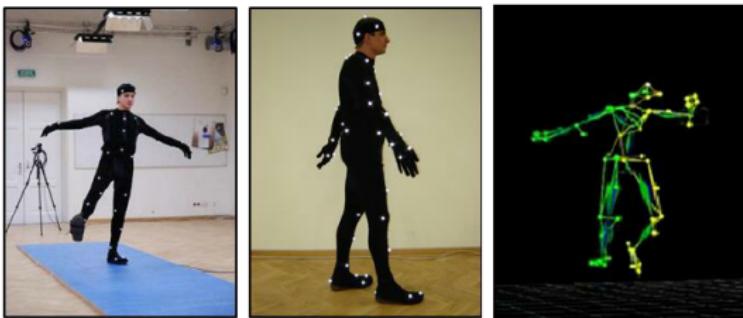
- Akgul, B. (2015). Door like parallelogram detection for robotic systems.
- Josinski, H., Kostrzewska, D., Michalczuk, A., and Switonski, A. (2014). The exiwo metaheuristic for solving continuous and discrete optimization problems.
- Muñoz-Salinas, R., Aguirre, E., García-Silvente, M., and González, A. (2004). Door-detection using computer vision and fuzzy logic. In *Proceedings of the 6th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering*.
- Riviere, V., Manecy, A., and Viollet, S. (2018). Agile robotic fliers: A morphing-based approach. *Soft Robotics*. PMID: 29846133.
- Sanket, N. J., Singh, C. D., Ganguly, K., Fermüller, C., and Aloimonos, Y. (2018). Gapflyt: Active vision based minimalist structure-less gap detection for quadrotor flight. *CoRR*, abs/1802.05330.
- Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

SIMULATION

- ▶ Virtual Robotics Experimentation Platform (VREP)
- ▶ Input/Output same form as real Crazyflie
- ▶ Noisy, low resolution camera
- ▶ Noise IMU
- ▶ Delay to mimic data down-link

MOTION CAPTURE

- ▶ VICON
- ▶ Set of cameras
- ▶ Track balls on desired object
- ▶ Provide orientation/position but only w/i tracked space
- ▶ Safety net: bad UAV orientation/path = shut down



Source: Josinski et al. (2014)

HOUGH TRANSFORM

- ▶ Edge detector
- ▶ Parametrize line:
- ▶ $r = x * \cos(\theta) + y * \sin(\theta)$
- ▶ For each edge point: determine all r, θ solving above,
increment table
- ▶ Maximums of table give lines majority of points agree on

