Obstacle Detection through Forward Movement

Qiuwei Shou and Alan Wu, CSCI-B657 Computer Vision, Indiana University

Motivation

- Obstacle detection and avoidance are essential capabilities for autonomous robots (land and air)
- While traditional multi-sensory approach addresses obstacle detection, the use of a single (monocular) camera enables a more affordable solution
- Computational requirements must be manageable by embedded controllers
- Project is inspired by [Mori and Scherer, 2013][1] paper

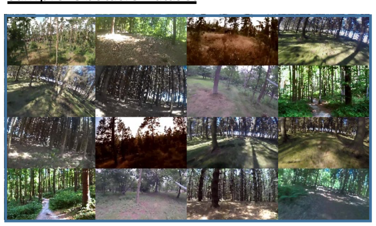
2

3

Dataset

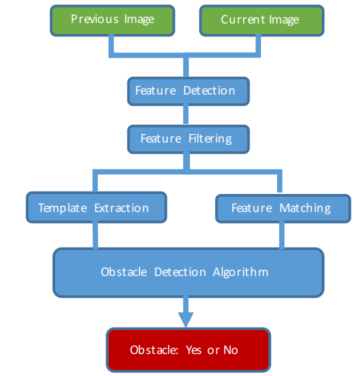
- We manually assembled our image dataset, which is primarily derived from drone hobbyist videos taken from a first-person view (FPV), downloaded from YouTube
- Mostly forest scenes with various levels of luminance and tilt angles throughout flight
- Two datasets totaling ~500 previous/current frame combinations
  - Set 1: ~400 dense forest
  - Set 2: ~100 open field (no obstacles)

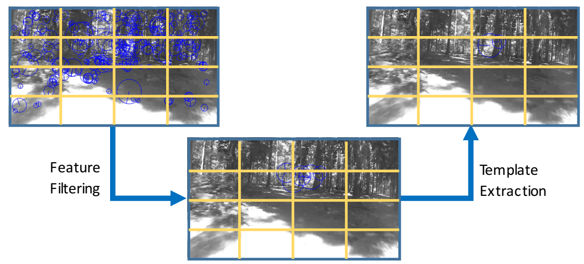
Sample Obstacle Dataset

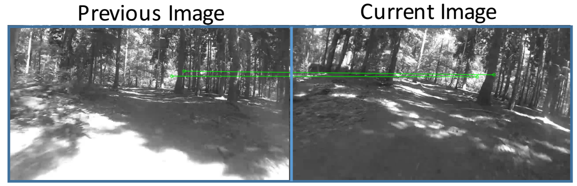


Sample Non-obstacle Dataset

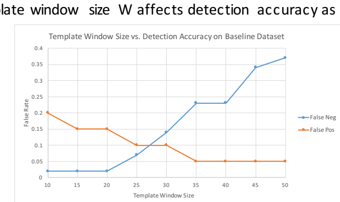
Methodology





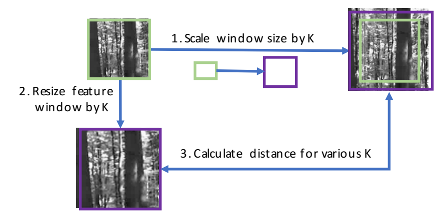


- We divide each image into 4x4 regions and use SURF for fast feature detection
- Only certain features in the "previous" frame are selected for matching with "current" frame
  - Features must be of a minimum size
  - Features must be within certain regions of the image
- We then perform template matching[1] to detect approaching object(s) that may be obstacles. See upper-right of poster for details
  - Initial template window size W affects detection accuracy as seen in graph below



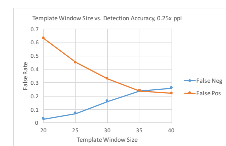
Template Matching

- Perform template matching[1] to determine scale increase of approaching object
  - For each remaining feature after filtering in "previous" frame, a small window is drawn around feature
  - A similar window is drawn around matching feature in "current" frame, with window size scaled up by K; resize window in "previous" frame also by K
  - Scalar K is determined such that the correlation distance (SSD) between "previous" and "current" windows is minimized. Discard any matches with distance $D > D_{max}$ and $K < K_{min}$



Results

- We achieved an accuracy of **84%** (i.e. False Neg = 16%) in detecting obstacles in Set 1 (dense forest) and a False Pos rate of **33%** in falsely detecting obstacles in Set 2 (open field), for initial window size of 30 and ¼ resolution relative to original (72 ppi)
- Plots show effects of *initial* window size W and image resolution



Challenges

- Dataset is largely "uncalibrated" – frame captures estimated to contain obstacles, but not guaranteed
- SURF is not optimal with noisy background
- Often unable to differentiate features on the ground from protruding objects (such as trees), especially when the camera is tilted downward

Conclusion and Future Work

- Able to detect oncoming obstacles using a low computation algorithm based on SURF features and template matching
- Access textural library (ex. tree bark) for more targeted detection and matching; possibly combine with Visual SLAM to achieve higher fidelity in estimating obstacle depth

References:

[1] Mori, Tomoyuki and Sebastian Scherer. "First Results in Detecting and Avoiding Frontal Obstacles from a Monocular Camera for Micro Unmanned Aerial Vehicles." ICRA, 2013.

[2] http://www.ingeniousingenuity.com

[3] http://clipartion.com/