Real-time Template Matching on a 32-bit Microcontroller

Audrow Nash ○ audrow@umich.edu

Introduction

- **Goals**: Perform real-time template matching on a microcontroller through scan matching. Evaluate various modes of template matching in terms of run-time and accuracy on a set of images.
- **Motivation**: A light-weight template matcher that runs on cheap hardware with a small footprint is desirable for robotics applications.
- **Applications**: In the *APRIL* lab this system will likely be applied to the following:
  - Magic robots and autonomous vehicles for person detection.
  - Unmanned aerial vehicles for monitoring landfill waste degradation.
  - Blimp for robot localization in GPS denied environments.

Background

- **Template matching**
  - Simple method for object recognition.
  - Find the location of the best match of a smaller template image on larger query image.
- **Scan matching**
  - Graph Based SLAM algorithm that recursively decimates, then starting at the smallest image, builds a search tree to quickly match images.
  - For closing the loop on 2D topographic maps using LiDAR sensors, Scan Matching increases computation speed by 45x against a flat image, [1].
  - *Process*
    1. Recursive decimation of template and query image.
    2. Compare smallest decimated template against corresponding query image.
    3. Use the best match to inform where to look in the next larger image.
    4. Repeat 2 and 3 until match is found or time expires.
- **Possible statistics between template and query image**
  - *Minimum error*: take the p-norm.
  - *Min-max*: compare minimum and maximum values.
  - *Mean interval*: compare the average value.
  - *Mutual information*: look for mutual dependence.

Approach

- **Camera board**
  - *ARM SAMD21*: ARM Cortex M0, 32kB RAM, 256kB ROM, 48MHz, operates at 1.6–3.2V.
  - *Lepton FLIR*, thermal camera: 80×60 pixel resolution, 63° diagonal field of view, 8–14 microns wavelength.
- **Metrics**
  - For various methods with a set of template and query images, compare the detection accuracy and run-time over numerous runs.
  - Average run-time and accuracy results to draw conclusions about behavior.
  - Record frequency that embedded platform returns accurate result 95% of the time.

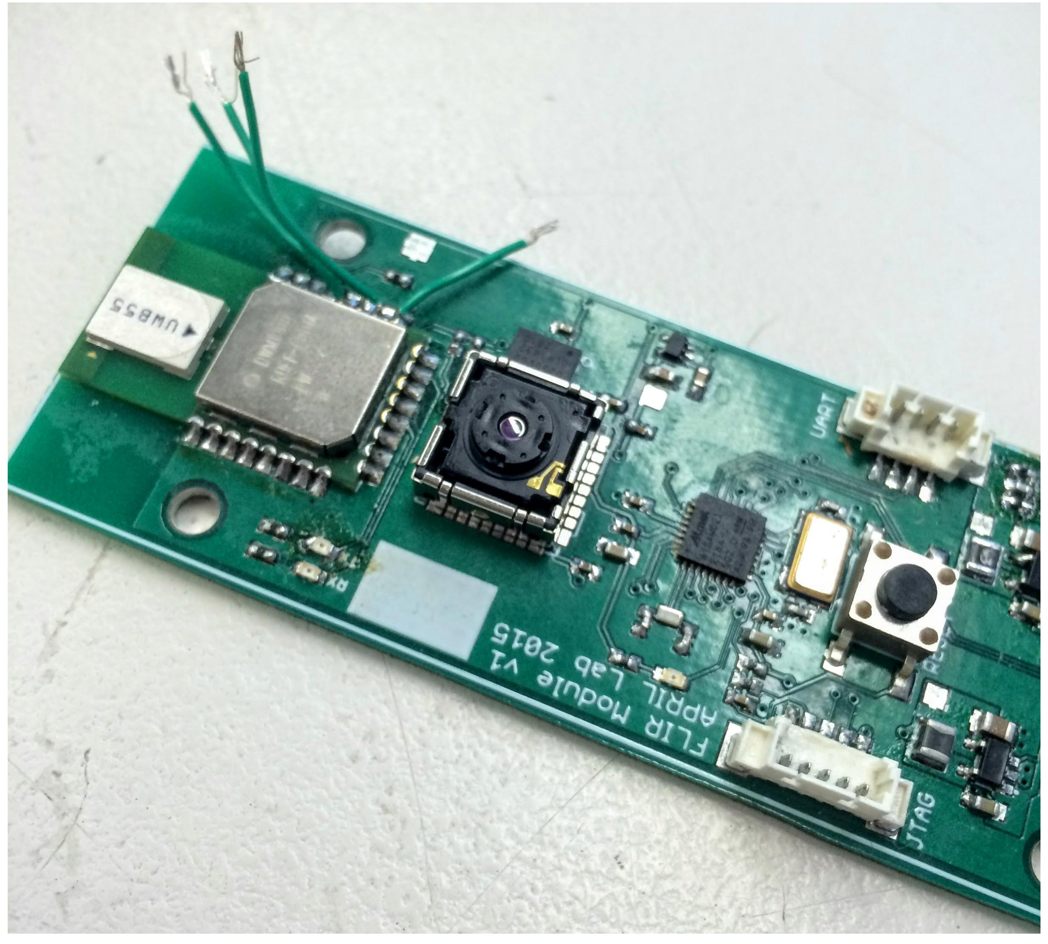

Fig. 1: Camera board developed by *APRIL* lab. In the center is the Lepton FLIR, thermal camera.

Progress

- Using images from the Lepton FLIR, built a person detector template through segmenting images, identifying person, and then averaging extracted person.

Fig. 2: Segmentation process for building template. Top left: original Lepton FLIR image. Top right, basic edge detection. Bottom left, subtract edge detection results from original image. Bottom right, binary threshold applied to image.

Fig. 3: Person detecting template generated through averaging several extracted person segments.

- Results using template on image outside of training set with minimum error template matching.

Fig. 4: Applying template to an image from the Lepton FLIR using minimum error template matching.

- Min-max statistic runs.
- Nearly finished implementing mean interval statistic with decimation.
- Implemented mutual information without decimation.
- Writing system calls to port code to embedded hardware.

Direction

1. Port min-max matching to SAMD21.
2. Finish implementation of mutual information and mean interval statistics.
3. Evaluate run-time and accuracy on and off the SAMD21.

References

[1] Edwin Olson. Real-Time Correlative Scan Matching. *Proceedings of the {IEEE} International Conference on Robotics and Automation*, pages 4387–4393, 2009.