

# METR4202: Advanced Robotics and Control

## Tutorial 8: SIFT and Image Features

---

In this tutorial we will investigate the question of how to find a known object within an image. Previous tutorials have shown examples of ‘dumb’ methods that look for e.g. circles or certain colors. Here we explore more robust methods that work using image ‘features’

### 1 Reading

Install the VLFeat toolbox for Matlab (from the course or VLFeat websites) and read through [the VLFeat SIFT tutorial](#).

Review the lecture slides on SIFT image features.

### 2 Template Matching

Imagine you have just landed on an alien planet. How would you describe an Australian \$50 note to a friendly alien (who happens to understand english)?

One way to find an object within an image is ‘template matching’. Template matching involves taking an existing image of an object (the template), then sweeping it over every pixel in a new image. At each pixel, a measure of the ‘difference’ between the two images is computed using some metric (e.g. absolute value of the difference). This difference then becomes the pixel value in an output image. Thus, at regions in the test image where the template matches, bright pixels will occur.

The Robotics Toolbox includes the `isimilarity` command to do just this. Use this command to template match the ‘fifty\_text.jpg’ template the first test case image (‘test\_1.jpg’). Note that `isimilarity` requires both image be grayscale, and the template image must have an odd width and height.

In the output image, the brightest pixels correspond to the best matches. Note that the edges are blacked out where the template was ‘hanging’ off the edge of the image slightly. Plot the original test image, and overlay a red ‘x’ for the best match found.

Try this again, but with the second test image. Are you able to find the template in the image? What are the limitations of a ‘dumb’ algorithm like template matching? How could this method be improved?

### 3 SIFT Feature Extraction

The previous section showed that template matching is not robust to image changes such as rotation, brightness and scale. A more robust way to match object images is using image features, as described in lectures. One very popular image feature algorithm is the Scale Invariant Feature Transform, or SIFT.

By following the VLFeat SIFT tutorial (on the VLFeat website), use the provided training images to build two SIFT databases of features for an AU\$50 note (one for the back, and one for the front). Plot the SIFT features and descriptors for one of these training images.

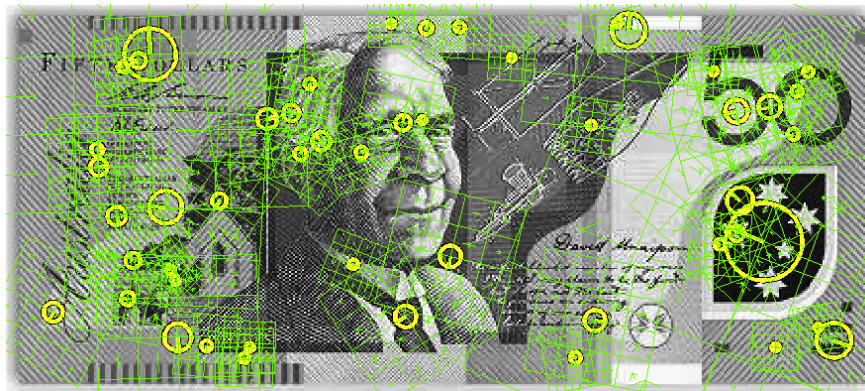


Figure 1: SIFT Features for the front of the \$50 note

### 4 SIFT Feature Matching

Try matching these descriptors with descriptors from the two testing images. The provided function `visualise_sift_matches` can be used to show the SIFT feature correlations between two images. Are you able to automatically find the \$50 note?

The VLFeat methods provide a 'score' for each match found. Change your code to only use the top 10 matches, or to use a score 'threshold'. Compare the benefits and limitations of these two approaches (thresholding vs taking the top N matches).

What limitations does a feature-based method like SIFT have?