

Feature Detection and Description

- **Understanding Features**

What are the main features in an image? How can finding those features be useful to us?

- **Harris Corner Detection**

Okay, Corners are good features? But how do we find them?

- **Shi-Tomasi Corner Detector & Good Features to Track**

We will look into Shi-Tomasi corner detection

- **Introduction to SIFT (Scale-Invariant Feature Transform)**

Harris corner detector is not good enough when scale of image changes. Lowe developed a breakthrough method to find scale-invariant features and it is called SIFT

- **Introduction to SURF (Speeded-Up Robust Features)**

SIFT is really good, but not fast enough, so people came up with a speeded-up version called SURF.

- **FAST Algorithm for Corner Detection**

All the above feature detection methods are good in some way. But they are not fast enough to work in real-time applications like SLAM. There comes the FAST algorithm, which is really "FAST".

- **BRIEF (Binary Robust Independent Elementary Features)**

SIFT uses a feature descriptor with 128 floating point numbers. Consider thousands of such features. It takes lots of memory and more time for matching. We can compress it to make it faster. But still we have to calculate it first. There comes BRIEF which gives the shortcut to find binary descriptors with less memory, faster matching, still higher recognition rate.

- **ORB (Oriented FAST and Rotated BRIEF)**

SIFT and SURF are good in what they do, but what if you have to pay a few dollars every year to use them in your applications? Yeah, they are patented!!! To solve that problem, OpenCV devs came up with a new "FREE" alternative to SIFT & SURF, and that is ORB.

- **Feature Matching**

We know a great deal about feature detectors and descriptors. It is time to learn how to match different descriptors. OpenCV provides two techniques, Brute-Force matcher and FLANN based matcher.

- **Feature Matching + Homography to find Objects**

Now we know about feature matching. Let's mix it up with calib3d module to find objects in a complex image.