

2D Features framework (feature2d module)

Learn about how to use the feature points detectors, descriptors and matching framework found inside OpenCV.

- **Harris corner detector**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

Why is it a good idea to track corners? We learn how to use the Harris method to detect corners.

- **Shi-Tomasi corner detector**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

Where we use an improved method to **detect corners more accurately**.

- **Creating your own corner detector**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

Here you will learn how to use OpenCV functions to make your personalized corner detector!

Languages: C++, Java, Python

- **Detecting corners location in subpixels**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

Is pixel resolution enough? Here we learn a simple method to improve our corner location accuracy.

- **Feature Detection**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

In this tutorial, you will use **features2d to detect interest points**.

- **Feature Description**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

In this tutorial, you will use **features2d to calculate feature vectors**.

- **Feature Matching with FLANN**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

In this tutorial, you will use the FLANN library to make a fast matching.

- **Features2D + Homography to find a known object**

Languages: C++, Java, Python

Compatibility: > OpenCV 2.0

Author: Ana Huamán

In this tutorial, you will use *features2d* and *calib3d* to detect an object in a scene.

- **Detection of planar objects**

Languages: C++

Compatibility: > OpenCV 2.0

Author: Victor Eruhimov

You will use *features2d* and *calib3d* modules for detecting known planar objects in scenes.

- **AKAZE local features matching**

Languages: C++, Java, Python

Compatibility: > OpenCV 3.0

Author: Fedor Morozov

Using AKAZE local features to find correspondence between two images.

- **AKAZE and ORB planar tracking**

Languages: C++

Compatibility: > OpenCV 3.0

Author: Fedor Morozov

Using AKAZE and ORB for planar object tracking.

- **Basic concepts of the homography explained with code**

Languages: C++, Java, Python

Compatibility: > OpenCV 3.0

This tutorial will explain the basic concepts of the homography with some demonstration codes.