|  |  |
| --- | --- |
| cv2.imread, cv2.imwrite | Reading and writing image from file |
| cv2.imshow | Displaying an image |
| cv2.waitKey(), cv2.waitKey(100) | Wait till user presses a key (for at most 100 msec) |
| cap.cvVideoCapture()  status, frame = cap.read() | Connecting to the camera and grabbing a frame |
| cv2.imresize(im, (width,height)) | Resizing an image |
| orb = cv2.ORB\_create()  keypoints, descriptors = orb.detectAndCompute(image, None) | Finding keypoints and computing them with descriptors |
| cv2.drawKeypoints | Display keypoints |
| bf = cv2.BFMatcher(cv.NORM\_HAMMING)  matches = bf.knnMatch(descr0,descr1,k=2) | Matching keypoints (two keypoints per one) by brute force |
| cv2.findHomography(src\_pts, dst\_pts, cv.RANSAC, tau) | Fit homography using RANSAC |
| from skimage import transform as tf  tf.warp(kitty, tf.ProjectiveTransform(M)) | Applying homography to an image ‘kitty’ |
| cap.release()  cv2.destroyAllWindows() | Correct termination of the program |

Most of stuff you need is here:

[**http://docs.opencv.org/3.0-beta/doc/py\_tutorials/py\_tutorials.html**](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html)

[**http://scikit-image.org/docs/dev/auto\_examples/applications/plot\_geometric.html**](http://scikit-image.org/docs/dev/auto_examples/applications/plot_geometric.html)