3.4

Required functions attached.
Observation: Homography keeps line straight, and my pictures can fit into the planar surfaces by Homography; while affine transformation keeps parallel lines, so my pictures can’t always fit into the planar surfaces, since some planar surfaces have non-parallel lines.

```
import numpy as np
from numpy.linalg import inv
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from scipy import ndimage
from PIL import Image

def affine_solve(U, V):
    b = np.ravel(V, order='F')
    b = b.reshape((b.shape[0], 1))
    A = get_A_affine(U[:, 0], V[:, 0])
    for i in range(1, U.shape[1]):
        A = np.vstack((A, get_A_affine(U[:, i], V[:, i])))
    h = inv(A.T @ A) @ A.T @ b
    H = np.vstack((h[:3, :].T, h[3:6, :].T, [[0, 0, 1]]))
    return H

def homography_solve(U, V):
    b = np.ravel(V, order='F')
    b = b.reshape((b.shape[0], 1))
    A = get_A_homo(U[:, 0], V[:, 0])
    for i in range(1, U.shape[1]):
        A = np.vstack((A, get_A_homo(U[:, i], V[:, i])))
    h = inv(A.T @ A) @ A.T @ b
    H = np.vstack((h[:3, :].T, h[3:6, :].T, [[h[6, 0], h[7, 0], 1]]))
    return H

def homography_transform(u, H):
    N = u.shape[1]
    u = np.vstack((u, np.ones((1, N))))
    V_target = H @ u
    V_target = from_3D_to_2D(V_target)
    V_target =  V_target.astype(int)
    return V_target


def get_A_homo(u, v):
    return np.array([[u[0], u[1], 1, 0, 0, 0, -u[0]*v[0], -u[1]*v[0]],  [0, 0, 0, u[0], u[1], 1, -u[0]*v[1], -u[1]*v[1]]])
```

```
def get_A_affine(u, v):
    return np.array([[u[0], u[1], 1, 0, 0, 0],  [0, 0, 0, u[0], u[1], 1]])

def from_3D_to_2D(V):
    N = V.shape[1]
    V = V @ inv(np.diag(V[2, 0:N]))
    return V[0:2, :]
```