

Detecting corners

Corner detection is an important process in Computer Vision. It helps us identify the salient points in the image. This was one of the earliest feature extraction techniques that was used to develop image analysis systems.

How to do it...

1. Create a new Python file, and import the following packages:

```
import sys

import cv2
import numpy as np
```

2. Load the input image. We will use [box.png](#) :

```
# Load input image -- 'box.png'
input_file = sys.argv[1]
img = cv2.imread(input_file)
cv2.imshow('Input image', img)
```

3. Convert the image to grayscale and cast it to floating point values. We need the floating point values for the corner detector to work:

```
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_gray = np.float32(img_gray)
```

4. Run the **Harris corner detector** function on the grayscale image. You can learn more about Harris corner detector at

http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html

```
# Harris corner detector
img_harris = cv2.cornerHarris(img_gray, 7, 5, 0.04)
```

5. In order to mark the corners, we need to dilate the image, as follows:

```
# Resultant image is dilated to mark the corners
img_harris = cv2.dilate(img_harris, None)
```

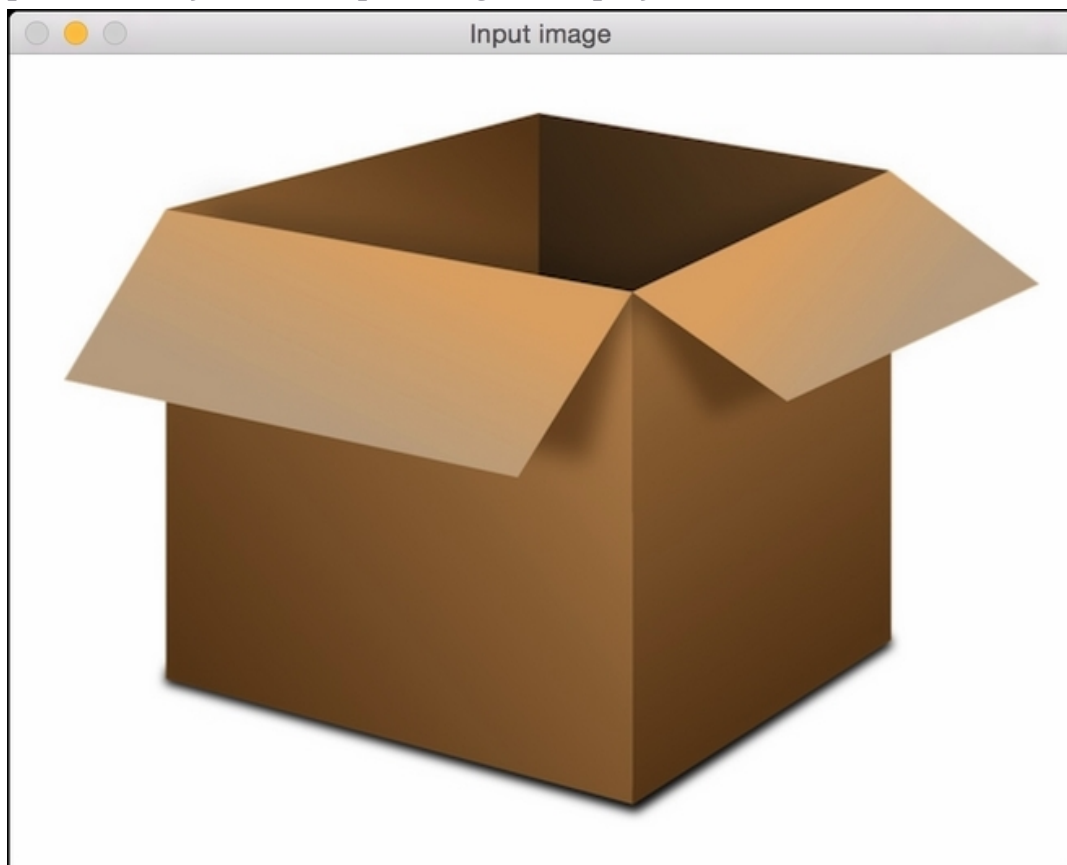
6. Let's threshold the image to display the important points:

```
# Threshold the image
img[img_harris > 0.01 * img_harris.max()] = [0, 0, 0]
```

7. Display the output image:

```
cv2.imshow('Harris Corners', img)
cv2.waitKey()
```

8. The full code is given in the [corner_detector.py](#) file that is already provided to you. The input image is displayed, as follows:



9. The output image after detecting corners is as follows:

