

1. preCornerDetector

In []:

```
import cv2
import numpy as np

def findLocalMaxima(src):
    kernel= cv2.getStructuringElement(shape=cv2.MORPH_RECT, ksize=(11,11))
    dilate = cv2.dilate(src, kernel)
    localMax = (src == dilate)

    erode = cv2.erode(src, kernel)
    localMax2 = (src > erode)
    localMax &= localMax2
    points = np.argwhere(localMax == True)
    points[:,[0, 1]] = points[:,[1, 0]] # switch x, y
    return points
```

In []:

```
src = cv2.imread('./data/CornerTest.jpg')
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
res = cv2.preCornerDetect(gray, ksize=3)
ret, res2 = cv2.threshold(np.abs(res), 0.1, 0, cv2.THRESH_TOZERO)
corners = findLocalMaxima(res2)
print(len(corners))

dst = src.copy()
for x, y in corners:
    cv2.circle(dst, (x, y), 5, (0,0,255), 2)

cv2.imshow('src', src)
cv2.imshow('dst', dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

2. cornerMinEigenVal

In []:



```
src = cv2.imread('./data/CornerTest.jpg')
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
eigen = cv2.cornerMinEigenVal(gray, blockSize=5)
print('eigen.shape=', eigen.shape)

T = 0.2
corners = np.argwhere(eigen > T)
corners[:, [0, 1]] = corners[:, [1, 0]] # switch x, y
print('len(corners) =', len(corners))
dst = src.copy()
for x, y in corners:
    cv2.circle(dst, (x, y), 3, (0,0,255), 2)

cv2.imshow('dst', dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

3. cornerHarris

In []:



```
src = cv2.imread('./data/CornerTest.jpg')
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
res = cv2.cornerHarris(gray, blockSize=5, ksize=3, k=0.01)
ret, res = cv2.threshold(np.abs(res), 0.02, 0, cv2.THRESH_TOZERO)
corners = findLocalMaxima(res)
print('corners=', corners)

dst = src.copy()
for x, y in corners:
    cv2.circle(dst, (x, y), 3, (0,0,255), 2)
cv2.imshow('dst', dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

4. FAST

In []:



```
src = cv2.imread('./data/CornerTest.jpg')
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)

fast = cv2.FastFeatureDetector_create(30)
fast.setNonmaxSuppression(0)
kp = fast.detect(src, None)
dst = src.copy()
cv2.drawKeypoints(src, kp, dst, (0, 0, 255))

cv2.imshow('dst', dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

5. ChessboardCorners

In []:



```
src = cv2.imread('./data/ChessBoard.jpg')
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
patternSize = (6, 3)
found, corners = cv2.findChessboardCorners(src, patternSize)
print('corners.shape=', corners.shape)

dst = src.copy()
cv2.drawChessboardCorners(dst, patternSize, corners, found)

cv2.imshow('dst', dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

6. CirclesGrid

In []:



```
src = cv2.imread('./data/circleGrid.jpg')
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
patternSize = (6, 4)
found, centers = cv2.findCirclesGrid(src, patternSize)
print('centers.shape=', centers.shape)

dst = src.copy()
cv2.drawChessboardCorners(dst, patternSize, centers, found)

cv2.imshow('dst', dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

In []:



