

## Matching using Hu Invariant Moment

In [ ]:



```
import cv2
import numpy as np

ref_src = cv2.imread('./data/refShapes.jpg')
ref_gray = cv2.cvtColor(ref_src, cv2.COLOR_BGR2GRAY)
ret, ref_bin = cv2.threshold(ref_gray, 0, 255,
                             cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)

test_src = cv2.imread('./data/testShapes.jpg')
test_gray = cv2.cvtColor(test_src, cv2.COLOR_BGR2GRAY)
ret, test_bin = cv2.threshold(test_gray, 0, 255,
                              cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)

mode = cv2.RETR_EXTERNAL
method = cv2.CHAIN_APPROX_SIMPLE
ref_contours, _ = cv2.findContours(ref_bin, mode, method)
test_contours, _ = cv2.findContours(test_bin, mode, method)

ref_dst = ref_src.copy()
colors = ((0,0,255), (0,255,0), (255,0,0))
for i, cnt in enumerate(ref_contours):
    cv2.drawContours(ref_dst, [cnt], 0, colors[i], 2)

test_dst = test_src.copy()
method = cv2.CONTOURS_MATCH_I1
for i, cnt1 in enumerate(test_contours):
    matches = []
    for cnt2 in ref_contours:
        ret = cv2.matchShapes(cnt1, cnt2, method, 0)
        matches.append(ret)
    k = np.argmin(matches)
    cv2.drawContours(test_dst, [cnt1], 0, colors[k], 2)

cv2.imshow('ref_dst', ref_dst)
cv2.imshow('test_dst', test_dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [ ]:

