CONTOUR 찾기

| import cv2  img\_src = cv2.imread('images/shape.png', cv2.IMREAD\_COLOR)  #1 - 이미지를 Gray로 변환  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  #2 이진화를 진행  # 이미지의 특성을 파악 : 검출할려고 하는것(도형)이 흰색으로 나와야함  # 배경이 흰색 : 검출해야하는 물체보다 배경이 밝은 상태  # 방법 2-1 : 그레이 이미지를 반전하고 Threshold 를 적용  # img\_gray = cv2.bitwise\_not(img\_gray)  # ret, img\_binary = cv2.threshold(img\_gray, 100, 255, cv2.THRESH\_BINARY)  # 방법 2-2 : cv2.threshold()함수의 옵션사용  ret, img\_binary = cv2.threshold(img\_gray, 150, 255, cv2.THRESH\_BINARY\_INV)  cv2.imshow('dst',img\_binary)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

컨투어 추출 및 그리기

| import cv2  img\_src = cv2.imread('images/shape.png', cv2.IMREAD\_COLOR)  #1 - 이미지를 Gray로 변환  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  #2 이진화를 진행  # 방법 2-2 : cv2.threshold()함수의 옵션사용  ret, img\_binary = cv2.threshold(img\_gray, 150, 255, cv2.THRESH\_BINARY\_INV)  # 검출하려고 하는 도형의 외곽선 검출 : findContours()함수 사용  contours, hierarchy = cv2.findContours(img\_binary, cv2.RETR\_CCOMP,  cv2.CHAIN\_APPROX\_NONE)  my\_color = (255,0,0)  thickness = 2  for i, contour in enumerate(contours):  cv2.drawContours(img\_src, [contour], 0, my\_color, thickness)  cv2.putText(img\_src, str(i), tuple(contour[0][0]),  cv2.FONT\_HERSHEY\_COMPLEX, 0.8, my\_color, 1)  cv2.imshow('dst',img\_src)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

drawContour사용법

| contours, hierarchy = cv2.findContours(img\_binary, cv2.RETR\_CCOMP,  cv2.CHAIN\_APPROX\_NONE)  my\_color = (255,0,0)  thickness = 2  # for문을 사용할 때 contour를 사용하는 방법  for i, **contour** in enumerate(contours):  cv2.drawContours(img\_src, **[contour], 0**, my\_color, thickness)  cv2.imshow('dst',img\_src)  cv2.waitKey(0)  # for문에서 drawContours함수의 인덱스를 사용하는 방법  for i, contour in enumerate(contours):  cv2.drawContours(img\_src, **contours, i**, my\_color, thickness)  cv2.imshow('dst',img\_src)  cv2.waitKey(0)  #외곽선만 표시할 경우, 한꺼번에 외곽선을 다 그림  cv2.drawContours(img\_src, **contours, -1**, my\_color, thickness)  cv2.imshow('dst',img\_src)  cv2.waitKey(0) |
| --- |

contourArea()를 사용하여 객체의 면적구하기

| import cv2  img\_src = cv2.imread('images/shape.png', cv2.IMREAD\_COLOR)  #1 - 이미지를 Gray로 변환  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  #2 이진화를 진행  # 이미지의 특성을 파악 : 검출할려고 하는것(도형)이 흰색으로 나와야함  # 배경이 흰색 : 검출해야하는 물체보다 배경이 밝은 상태  # 방법 2-1 : 그레이 이미지를 반전하고 Threshold 를 적용  # img\_gray = cv2.bitwise\_not(img\_gray)  # ret, img\_binary = cv2.threshold(img\_gray, 100, 255, cv2.THRESH\_BINARY)  # 방법 2-2 : cv2.threshold()함수의 옵션사용  ret, img\_binary = cv2.threshold(img\_gray, 150, 255, cv2.THRESH\_BINARY\_INV)  # 검출하려고 하는 도형의 외곽선 검출 : findContours()함수 사용  contours, hierarchy = cv2.findContours(img\_binary, cv2.RETR\_LIST,  cv2.CHAIN\_APPROX\_NONE)  my\_color = (0,255,0) # (B,G,R)  thickness = 2  for i, contour in enumerate(contours):  area = cv2.contourArea(contour)  # contourArea()함수로 객체의 면적을 구하고 면적기준으로 임계값보다  # 큰 객체만 외곽선을 그리고 면적정보를 표시한다.  if area > 10000:  cv2.drawContours(img\_src, contours, i, my\_color, thickness)  cv2.putText(img\_src, f'{i}: {int(area)}', tuple(contour[0][0]),  cv2.FONT\_HERSHEY\_COMPLEX, 0.8, my\_color, 1)  cv2.imshow('dst',img\_src)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

모멘트 구해서 표시하기

| import cv2  img\_src = cv2.imread('images/shape.png', cv2.IMREAD\_COLOR)  #1 - 이미지를 Gray로 변환  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  #2 이진화를 진행  # 이미지의 특성을 파악 : 검출할려고 하는것(도형)이 흰색으로 나와야함  # 배경이 흰색 : 검출해야하는 물체보다 배경이 밝은 상태  # 방법 2-1 : 그레이 이미지를 반전하고 Threshold 를 적용  # img\_gray = cv2.bitwise\_not(img\_gray)  # ret, img\_binary = cv2.threshold(img\_gray, 100, 255, cv2.THRESH\_BINARY)  # 방법 2-2 : cv2.threshold()함수의 옵션사용  ret, img\_binary = cv2.threshold(img\_gray, 150, 255, cv2.THRESH\_BINARY\_INV)  # 검출하려고 하는 도형의 외곽선 검출 : findContours()함수 사용  contours, hierarchy = cv2.findContours(img\_binary, cv2.RETR\_LIST,  cv2.CHAIN\_APPROX\_NONE)  my\_color = (0,255,0) # (B,G,R)  text\_color = (255,0,0) # (B,G,R)  thickness = 2  for i, contour in enumerate(contours):  area = cv2.contourArea(contour)  # contourArea()함수로 객체의 면적을 구하고 면적기준으로 임계값보다  # 큰 객체만 외곽선을 그리고 면적정보를 표시한다.  if area > 10000:  cv2.drawContours(img\_src, contours, i, my\_color, thickness)  #모멘트 그리기(무게중심)  mu = cv2.moments(contour)  cx = int(mu['m10'] / (mu['m00']+1e-5))  cy = int(mu['m01'] / (mu['m00']+1e-5))  cv2.circle(img\_src, (cx,cy), 5, (0,255,255),-1)  cv2.putText(img\_src, f'{i}: {int(area)}', (cx-50,cy-20),  cv2.FONT\_HERSHEY\_COMPLEX, 0.8, text\_color, 1)  cv2.imshow('dst',img\_src)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

객체의 외곽을 구성하는 사각형 그리기

| import cv2  import numpy as np  img\_src = cv2.imread('images/shape.png', cv2.IMREAD\_COLOR)  #1 - 이미지를 Gray로 변환  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  #2 이진화를 진행  # 이미지의 특성을 파악 : 검출할려고 하는것(도형)이 흰색으로 나와야함  # 배경이 흰색 : 검출해야하는 물체보다 배경이 밝은 상태  # 방법 2-1 : 그레이 이미지를 반전하고 Threshold 를 적용  # img\_gray = cv2.bitwise\_not(img\_gray)  # ret, img\_binary = cv2.threshold(img\_gray, 100, 255, cv2.THRESH\_BINARY)  # 방법 2-2 : cv2.threshold()함수의 옵션사용  ret, img\_binary = cv2.threshold(img\_gray, 150, 255, cv2.THRESH\_BINARY\_INV)  # 검출하려고 하는 도형의 외곽선 검출 : findContours()함수 사용  contours, hierarchy = cv2.findContours(img\_binary, cv2.RETR\_LIST,  cv2.CHAIN\_APPROX\_NONE)  my\_color = (0,255,0) # (B,G,R)  text\_color = (255,0,0) # (B,G,R)  thickness = 2  for i, contour in enumerate(contours):  area = cv2.contourArea(contour)  # contourArea()함수로 객체의 면적을 구하고 면적기준으로 임계값보다  # 큰 객체만 외곽선을 그리고 면적정보를 표시한다.  if area > 10000:  cv2.drawContours(img\_src, contours, i, my\_color, thickness)  #모멘트 그리기(무게중심)  mu = cv2.moments(contour)  cx = int(mu['m10'] / (mu['m00']+1e-5))  cy = int(mu['m01'] / (mu['m00']+1e-5))  cv2.circle(img\_src, (cx,cy), 5, (0,255,255),-1)  cv2.putText(img\_src, f'{i}: {int(area)}', (cx-50,cy-20),  cv2.FONT\_HERSHEY\_COMPLEX, 0.8, text\_color, 1)  # 객체의 외곽에 사각형 그리기  # 방법 1: boundingRect(회전 고려 않함) : Cyan  x,y,w,h = cv2.boundingRect(contour)  cv2.rectangle(img\_src, (x,y), (x+w,y+h),(255,255,0), 1)  # 방법 2: minAreaRect() 사용 : Magenta  # 물체의 회전을 고려해서 경계 사각형 그림  rect = cv2.minAreaRect(contour)  box = cv2.boxPoints(rect)  box = np.int0(box)  cv2.drawContours(img\_src, [box], 0, (255,0,255), 1)  cv2.imshow('dst',img\_src)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

## 

## 모폴로지 연산

형태소(Structing element)의 크기와 형태를 지정

크기 : 3x3 5x5 7x7 9x9

형태 : 십자가모양(cv2.MORPH\_CROSS), 직사각형(cv2.MORPH\_RECT), 타원(cv2.MORPH\_ELLIPSE)

kernel = cv2.getStucturingElement( 형태, 형태소크기(9,9))

Erosion 코드

| import cv2  img\_src = cv2.imread('images/shape.png', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src,cv2.COLOR\_BGR2GRAY)  img\_gray = cv2.bitwise\_not(img\_gray)  kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (7,7))  img\_erode = cv2.erode(img\_gray, kernel, iterations=3)  img\_dst = cv2.hconcat([img\_gray,img\_erode])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

Dilation 코드

| import cv2  img\_src = cv2.imread('images/shape.png', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src,cv2.COLOR\_BGR2GRAY)  img\_gray = cv2.bitwise\_not(img\_gray)  kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (7,7))  img\_dilate = cv2.dilate(img\_gray, kernel, iterations=3)  img\_dst = cv2.hconcat([img\_gray,img\_dilate])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

OPENING : erode후 dilate진행

| import cv2  img\_src = cv2.imread('images/shape-3.png', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src,cv2.COLOR\_BGR2GRAY)  img\_gray = cv2.bitwise\_not(img\_gray)  img\_morp = img\_gray.copy()  kernel = cv2.getStructuringElement(cv2.MORPH\_CROSS, (7,7))  # OPENING 5번은 erode 5번 진행후 dilate 5번 진행하는것과 같다.  img\_morp1 = cv2.morphologyEx(img\_morp, cv2.MORPH\_OPEN,  kernel, iterations=5)  img\_morp2 = cv2.erode(img\_morp, kernel, iterations=5)  img\_morp2 = cv2.dilate(img\_morp2, kernel, iterations=5)  img\_dst = cv2.hconcat([img\_morp1,img\_morp2])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| --- |

CLOSING : dilate 후 erode 진행

| import cv2  img\_src = cv2.imread('images/shape-3.png', cv2.IMREAD\_COLOR)  img\_gray = cv2.cvtColor(img\_src,cv2.COLOR\_BGR2GRAY)  img\_gray = cv2.bitwise\_not(img\_gray)  img\_morp = img\_gray.copy()  kernel = cv2.getStructuringElement(cv2.MORPH\_CROSS, (7,7))  # CLOSING 5번은 dilate5번 진행후 erode5번 진행하는것과 같다.  img\_morp1 = cv2.morphologyEx(img\_morp, cv2.MORPH\_CLOSE,  kernel, iterations=5)  img\_morp2 = cv2.dilate(img\_morp, kernel, iterations=5)  img\_morp2 = cv2.erode(img\_morp2, kernel, iterations=5)  img\_dst = cv2.hconcat([img\_morp1,img\_morp2])  cv2.imshow('dst',img\_dst)  cv2.waitKey(0) |
| --- |

원찾기

| import cv2  img\_src = cv2.imread('images/balls.jpg',cv2.IMREAD\_COLOR)  img\_dst = img\_src.copy()  img\_gray = cv2.cvtColor(img\_src, cv2.COLOR\_BGR2GRAY)  circles = cv2.HoughCircles(img\_gray, cv2.HOUGH\_GRADIENT, 1,  minDist=100, param1=250, param2=10,  minRadius=80, maxRadius=120)  for i, circle in enumerate(circles[0]):  # 값이 실수로 들어오므로 정수로 변환하여야 표시가 됨  cv2.circle(img\_dst,  (int(circle[0]),int(circle[1])), int(circle[2]),  (255,255,255), 3)  cv2.imshow('dst',img\_dst)  cv2.waitKey(0) |
| --- |