|  |  |
| --- | --- |
| 실습 13 (파노라마 이미지) | 소스코드 및 결과이미지  import cv2  import numpy as np  # 이미지 파일 경로 설정  image1 = "book1.jpg"  image2 = "book2.jpg"  # 이미지 불러오기  img1 = cv2.imread(image1)  img2 = cv2.imread(image2)  if img1 is None or img2 is None:      print("이미지를 불러오는데 실패했습니다.")  else:      # SIFT (Scale-Invariant Feature Transform) 특징점 검출기 초기화      sift = cv2.SIFT\_create()      # 특징점과 특징점 디스크립터를 찾음      kp1, des1 = sift.detectAndCompute(img1, None)      kp2, des2 = sift.detectAndCompute(img2, None)      # Brute-Force 기반 매칭      bf = cv2.BFMatcher()      matches = bf.knnMatch(des1, des2, k=2)      # 좋은 매치 선택      good\_matches = []      for m, n in matches:          if m.distance < 0.75 \* n.distance:              good\_matches.append(m)      # 좋은 매치를 사용하여 호모그래피 계산      if len(good\_matches) > 10:          src\_pts = np.float32([kp1[m.queryIdx].pt for m in good\_matches]).reshape(-1, 1, 2)          dst\_pts = np.float32([kp2[m.trainIdx].pt for m in good\_matches]).reshape(-1, 1, 2)          # 호모그래피 계산          H, \_ = cv2.findHomography(src\_pts, dst\_pts, cv2.RANSAC, 5.0)          # 이미지 이어붙이기          panorama = cv2.warpPerspective(img1, H, (img1.shape[1] + img2.shape[1], img1.shape[0]))          panorama[:img2.shape[0], :img2.shape[1]] = img2  # 결과 이미지 저장          cv2.imwrite("panorama.jpg", panorama)          # 결과 이미지 표시          cv2.namedWindow("Panorama", cv2.WINDOW\_NORMAL)          cv2.resizeWindow("Panorama", 800, 600)          cv2.imshow("Panorama", panorama)          cv2.waitKey(0)          cv2.destroyAllWindows()      else:          print("충분한 좋은 매치를 찾을 수 없습니다.") |
| 실습 14 (사람검출) | 소스코드 및 결과이미지  import cv2  # 동영상 파일 경로 설정  video\_path = "vtest.avi"  # HOG 디텍터 초기화  hog = cv2.HOGDescriptor()  hog.setSVMDetector(cv2.HOGDescriptor\_getDefaultPeopleDetector())  # 동영상 열기  cap = cv2.VideoCapture(video\_path)  while True:      ret, frame = cap.read()        if not ret:          break      # 이미지 크기를 조절 (선택적)      frame = cv2.resize(frame, (640, 480))        # HOG 기반 사람 검출      (rects, weights) = hog.detectMultiScale(frame, winStride=(4, 4), padding=(8, 8), scale=1.05)        # 검출된 사람 주위에 사각형 그리기      for (x, y, w, h) in rects:          cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)        # 결과 동영상 표시      cv2.imshow("Human Detection", frame)        if cv2.waitKey(1) & 0xFF == 27:  # 'ESC' 키로 종료          break  cap.release()  cv2.destroyAllWindows() |