|  |  |
| --- | --- |
| 교육 제목 | **AI 프로젝트 기반 S/W 전문가 양성 과정** |
| 교육 일시 | 21. 12. 21. |
| 교육 장소 | 영우글로벌러닝 |
| 교육생 | 박건준 |
| **교육 내용** | |
| 오전 | 몰폴로지 복습. = 팽창과 침식(클로징, 닫힘연산) = 4연결성 등의 알고리즘을 사용. 연산을 여러 번하고, 반대로 적용도 가능  HSV 혹은 HSI = 명도 채도를 선택 할 수 있음.  와이씨비씨알 YCbCr 모델 영상 및 비디오 압축  RGB  CMR  Geometic Transforms  화소의 값 연산이 아닌 픽셀 위치를 바꿔주는 것으로 이동, 회전. 대칭등   1. Spatial transform 좌표변환 = 변환된 x는 엑스 프라임 x’. 2. Gray-level interpolation 인터폴레이션 크기 줄일 때 각지지 않게 했던것   source 소스를 destination 결과물로 forward mapping = 2배로 키움.  그러나 픽셀에 hole이 발생하여 잘 안 씀. 소수점이 있으면 안되기 때문.  backward mapping 은 나누기 때문에 홀이 발생하지 않음. 그러나 정수가 아닌 소수점 형태 float의 좌표값이 나옴. 그래서 인터폴레이션을 함.  카운터클러와이즈  아규머트라는 플래그  Affine transform 어파인 트렌스포메이션 = 마주보는 대각이 항상 평행.  스케일, 회전, 스큐어리 skew  미니어 트렌스포메이션.  메트릭스 로테이션 : 대문자 A, 벡터라는 뜻의 x 위에 화살표표시 비슷한 것을 넣음.  homogeneous coordinate system 사용. 좌표계 행열 로테이션으로 바꿈.  더하기는 연산 할 때 불리하므로 메트릭스로 합쳐 버림.  어파인과는 구별 되는  워핑 warpping = 넌닝이라고 불리며 물결처럼 휘어지게 하는 것.  모핑 = 서로다른 사진이나 영상 등이 서서히 바뀌는 것.  코코싸싸  싸코코싸  ‘0에 / 있는 모양’ = 화이  쎄타는 0 모양 안에 가로줄  트렌스로 이동 후, 로테이션 메트릭스로 회전, 다시 트렌스로 이동.  영상 중심점만큼 이동하고, 회전 후 다시 제자리로 돌려놓음.  단 3차원은 로우가 더 붙어서 복잡함.  위의 순서대로 할 것을 1개의 매트릭스로 만들 수 있음.  스케일링 : 확대축소 : S 상수로 몇배 곱할지를 정함.  스큐어리 skew는 Y값이 증가할 수록 밀림 :  스큐 혹은 쉐어링이라고도 함. u를 씀  좌우 대칭 = 교재 오류  x’ = -x + widh-1  y’ = y + 0  상하 대칭  x’ = x + 0  y’ = -y, h-1  와핑 warping 넌리니어 트렌스폼 Nonlinear transform = 고차항을 사용.  인터폴레이셔 interpolation 소수점 값으로 된 좌표값을 주변 픽셀을 고려하여 새로운 값을 생성하는 방법.  Nearest neighbor interpolation 모자이크 영상 할 때 씀. 최근접 보강.  Neighbor averaging  Bilinear interpolation 네 개의 가장 가까운 픽셀들에 가중치를 곱한 값을 합함. 화소당 선형 보간을 세번씩 수행하므로 연산량이 많음. 영상도 볼만하고 연산도 적당함.  고차보간법 = B 스플라인 = 4x4 이웃 화소 참조. 3차원 보간법.  블러링을 먼저하면 축소 시에도 사라지는 것을 보완 할 수 있음.  안티 에일리어싱 Aliasing.  인터 에일리어 AA 쓰던 것 |
| 오후  (2시) | 복습  import sys  import cv2  import numpy as np  import matplotlib as plt  마스크로 이미지 넣기.  cv2.imread()로 함수안에 파일명과 확장자를 넣어서 파일 불러오고, 맨 좌측에 변수 이름을 만듬. 변수는 src, mask, dst  cv2.copyTo(src, mask, dst)로 합쳐버림.  cv2.imshow('dst', dst)로 결과 확인.  cv2.waitKey()  cv2.destroyAllWindows()  # 마스크 만들기 트렌스 홀드  retval, mask = cv2.threshold(img1\_gray, 240, 255, cv2.THRESH\_BINARY\_INV)  img1\_gray = cv2.cvtColor(img1, cv2.COLOR\_BGR2GRAY)  h, w = img1.shape[:2]  img2 = cv2.resize(img2, (w, h), cv2.INTER\_AREA)  # 히스토그램 조작  src = cv2.imread('fig/lenna.bmp', 0)  # cv2.imshow('src', src)  dst=cv2.add(src, 50)  hist = cv2.calcHist([dst], [0], None, [256], [0, 256])  #빈의 계수를 몇개로 할 것이냐. 영상에서는 0~255라서 256개임.  plt.plot(hist)  # plt.plot(src1)  # plt.axis('off')  plt.show()  # 히스토그램 스트레칭과 이퀄라이제이션  cv2.waitKey()  cv2.destroyAllWindows()  ## 트랙바를 이용해서 색상 찾기  import sys  import numpy as np  import cv2  src = cv2.imread('candies.png')  if src is None:  print('Image load failed!')  sys.exit()  src\_hsv = cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)  def on\_trackbar(pos):  hmin = cv2.getTrackbarPos('H\_min', 'dst')  hmax = cv2.getTrackbarPos('H\_max', 'dst')  # sat = cv2.getTrackbarPos('sat', 'dst')    dst = cv2.inRange(src\_hsv, (hmin, 150, 0), (hmax, 255, 255))  # dst = cv2.inRange(src\_hsv, (hmin, sat, 0), (hmax, 255, 255))  cv2.imshow('dst', dst)  cv2.imshow('src', src)  cv2.namedWindow('dst')  cv2.createTrackbar('H\_min', 'dst', 50, 179, on\_trackbar)  cv2.createTrackbar('H\_max', 'dst', 80, 179, on\_trackbar)  # cv2.createTrackbar('sat', 'dst', 150, 255, call\_trackbar)  on\_trackbar(0)  cv2.waitKey()  cv2.destroyAllWindows() |