|  |  |
| --- | --- |
| 교육 제목 | **Segmentation, Morphology,** **Geometric Transforms,마우스-키보드처리** |
| 교육 일시 | 2021.11.09 |
| 교육 장소 | C6 |
| **교육 내용** | |
|  | 복습:  Equalization: 히스토그램 분포 조절 (노이즈 발생)>>> <-> normalization  Uniform: 모든 확률이 동일  CDF: 누적 분포함수  Bimodal histogram: 봉우리가 2개 / 커팅라인을 어떻게 찾을 것이냐 -OTSU,LDA  Immersion simulation: 물을 뿌려 – 고이는 곳 봐라  **\*Image Segmentation:**  **전체 영상을 객체 또는 의미 있는 객체의 일부 영역의 집합으로 구분하는 과정**  **-군집화 기법** (Clustering Technique)=K-mean clustering을 영상 도메인에서 사용  -특징 공간(Feature space)에서 군집화(clustering)에 의해 영상분할(segmentation)  **-반복 임계화 기법**(Recursive Thresholding Technique) : 반복적으로 잘라 나간다  =iterative  **-연결 요소 레이블링**(Connected Component Labeling)  **\*Morphology(형태학): 픽셀의 모양을 다룬다**  Dilation(팽창)과 Erosion(침식) 무엇을 먼저 하느냐 에 따라 Opening과 Closing  침->팽 팽->침  -팽창연산: 4연결성(connectivity)=anchor point / 데이터가 겹치면 or 연산 수행  -침식연산: 객체 크기의 축소 , 작은 경계를 침식, 작은 돌기를 제거  -열림 연산(Opening operation)  -침식(Erosion) 연산을 수행한 후 다시 팽창(Dilation) 연산 적용  -작은 크기의 객체에 포함되는 픽셀들을 제거  -닫힘 연산(Closing operation)  -팽창(Dilation) 연산을 수행한 후 다시 침식(Erosion) 연산 적용  -객체 내부의 작은 구멍hole이나 간격gap을 채움  \***Geometric Transforms**: 수식이나 변환 관계에 의해 픽셀들의 위치를 변환,  이미지를 좌우로 변환, 회전  **-Spatial Transform(공간변환):**    **-forward mapping** (직접 돌림)  변환 수식에 의해 입력좌표를 출력좌표로 변환하는 과정  출력 영상에서 정의되지 않은 픽셀(hole) 발생  **-backward mapping** (얻고자 하는 영상을 놓고 반대로 매핑,역방향 매핑 =홀 x)  출력 영상의 각 픽셀 좌표에 대응하는 원본 영상의 좌표를 계산하여 해당 픽셀의 밝기 값을 결정하는 방법  출력 영상에서 정의되지 않은 픽셀 발생 방지  계산된 좌표가 정수가 아닌 경우 발생 → interpolation 적용  **-Affine transform** (=Linear transform = rigid)  휘어짐이 없고 평행한 선들은 평행을 유지하는 변환  이동, 회전, 스케일 및 이들의 조합에 의한 변환  -------------------Matrix 사용----------------------  = x쪽으로 2배  **-Warping** (Nonlinear transform (rubber sheet transform) )  -pixel별로 이동 정도를 다르게 할 수 있어서 고무 판 위에 그려진 영상을 임의대로 구부리는 것과 같은 효과를 낼 수 있음  - 이미지 굴곡이 생김  -고차항을 사용하여 일반화된 다항식으로 표현  이차항은 일정한 방향 굴곡 ,고차항은 굴곡이 요란해짐-> regulization 잘 해야함  **Interpolation**(보간 법/ 이미지를 가져올 때 )  -결과 픽셀에 정확하게 대응되는 입력 픽셀이 없는 경우 주변 픽셀들  을 고려하여 새로운 값을 생성하는 방법  **-Interpolation 종류**  **-Nearest neighbor interpolation**  계산한 위치에서 가장 가까운 원시 픽셀을 선택하는 방 법  처리 속도는 빠르지만 결과 영상의 질이 좋지 않음  **- Neighbor averaging interpolation**(다 더해서 나누자)  **- Bilinear interpolation**(많이 씀)  새로운 픽셀을 생성하기 위해 네 개의 가장 가까운 픽셀들에 가중치를 곱한 값들의 합을 사 (거리 합)  보다 자연스러운 영상을 산출  **- Higher order interpolation** (spline ,cubic?) |
|  | **마우스 이벤트 처리**    **Trackbar 사용**    **실습: 알파채널 이미지 불러오기**    **영상 더하기**    **dst = src+100** =밝기 높임 cv2.imshow k\*src +i  밝은곳에 100더하면 너무 밝음  해결방법  **dst=cv2.add(src,100)** or **dst=np.clip(src+100.,0,255).astype(np.uint8)**  연산은 float데이터 타입은 int로 바꿔야함    **dst = cv2.add(src,100)** # BGR B에만 밝기가 더 해짐  해결방법  **dst = cv2.add(src,(100,100,100,0))** #컬러색우지하고 밝기만/ 좋은 방법은 아님 |