|  |  |
| --- | --- |
| 교육 제목 | **HISTOGRAM, Equalization**  **OpenCV (그리기 함수, 카메라와 동영상 처리)** |
| 교육 일시 | 2021.11.08 |
| 교육 장소 | 비대면 |
| **교육 내용** | |
|  | 복습: pattern recognition: 영상 개선 -> 이미지 분할 -> 색상,모양정보->분류  영상복원 영상,**왜곡** (이유를 안다,렌즈 굴 곡률) **개선**(왜 그런지 잘 모름) **압축**  **HISTOGRAM: 픽셀 수를 제공하는 간단한 데이터 이미지에서 주어진 값.**  HISTOGRAM MODIFICATIONS(수정):  -range조정(stretching)  -shape 조정(Equalization) 눈으로 보기 좋은 방법 =히스토그램의 모양이 변함  **Stretching**:  shape 변하지 않고 대역만 증가  **Equalization**  \*CDF(Cumulative Distribution Function): 누적 분포 함수 = 더하면1이된다  -입력영상의 히스토그램의 값을 누적 시켜 각 레벨에서의 히스토그램 누적 합 계산  -히스토그램의 누적 합을 전체 픽셀의 개수로 나누어 값을 정규화 함  -정규화 된 값에 최대 gray level 값을 곱한 후 반올림을 수행  -입력 영상의 각 gray level에 대해 변환 값으로 대응  **GRAY-LEVEL THRESHOLDING(역치) =** binary =mask =이진영상  Value 기준으로 이미지를 나눈다  Binarization  사람손으로 하는 방법 = Intensity (밝기)만가지고 하는 건 좋은 방법이 아니다  OPTIMAL(이상적인) THRESHOLD BY OTSU  기본 원리  -임계 값 T를 기준으로 영역을 2개 그룹으로 나누었을 때 각 집합내 의 명암 분포는 균일하고 집합 사이의 명암 차이는 최대화될 수 있도록 함  -모든 가능한 T에 대해 점수를 계산하여 가장 좋은 T를 최종 임계 값으로 선택함  =최적화 알고리즘 (optimization algorithm)  낱낱 탐색 (exhaustive search), 언덕 오르기 (hill climbing) 등의 탐색 방법을 사용 가능  -최적화 알고리즘에서는 비용 함수 (cost function) 또는 목적 함수(objective function)을 사용하여 점수 계  **Watershed algorithm**  Watershed: 분수령  Catchment basin: 集水 구역 동일 호수로 물이 모이는 구역  Minina : 집수 구역의 최저점  Convolution(중첩적분)  -전통적인 알고리즘Conventional watershed algorithm  그라디언트 영상(gradient image)에 적용  -수정된 알고리즘Modified watershed algorithm  밝기 영상(intensity image)에 적용 가능  **Immersion(범람) simulation/ 옛날에 썼던 방법**  1. 에지 강도 영상을 생성  2. 지역적으로 최소값 픽셀들을 찾아 서로 다른 레이블 할당  3. 레이블 된 픽셀에 인접하면서 레이블 되지 않거나 watershed가 아닌 픽셀들을 찾아 큐에 추가  4. 큐에서 꺼낸 픽셀이 하나의 레이블에 인접하면 그 레이블을 할당하고 아니면 watershed로 설정  5. 큐에 남은 픽셀이 존재하지 않을 때까지 단계 3과 4를 반복 |
|  | **OpenCV 그리기**    **그리기 함수**  -라인 그리기  cv2.line(img, pt1, pt2, color, thickness = None, lineType = None, shift = None) -> img  -사각형 그리기  cv2.rectangle(img, pt1, pt2, color, thickness = None, lineType = None) -> img  pt1 :좌측 상단, pt2: 우측하단  cv2.rectangle(img, rect, color, thickness = None, lineType = None) -> img  rect: 사각형의 위치 정보 (x, y, w, h)  -원형 그리기  cv2.circle(img, center, radius, color, thickness = None, lineType = None) -> img  # center: 원의 중심좌표 (x, y) # radius : 원의 반지름  -다각형 그리기  cv2.polylines(img, pts, isClosed, color, thickness = None, lineType = None) -> img  # center: 다각형 점들의 array# isClosed : True for 폐곡선  **실습 – 인삼 쓰기**    **카메라와 동영상 처리** |