Data Science

데이터 분석 과정 5-2

2021년 1월 11일 (4일간)

김성우
(libero@deu.ac.kr)

내용

- 영상 분석 개요
- OpenCV 개요
- 필터링과 기하학적 변환
- 영상 신호 처리 개요
- 영상 분할
- 이진영상처리
- 영상 인식, 분류, 검출의 활용

영상 분석 개요

영상 분석 개요

- 영상 분석
  - 컴퓨터를 사용하여 영상 데이터를 분석하여 중요한 의미를 추출하거나 미시각 기능을 가진 장치을 만드는 기술

- 분야
  - 영상 처리 (Processing)
  - 검출 (Detection)
  - 분류 (Classification)
  - 인식 (Recognition)

영상 처리

- 영상을 입력으로 받아 처리하여 새로운 영상을 출력

- 필터링
- 이진화
- 모폴로지
- 기하학적 변환
- 영상 변환
- 영상 분할

컴퓨터 비전 처리 과정

- **전처리**
  - 주로 영상 처리

- **특징 추출**
  - 에지, 선분, 영역, 텍스처, 지역 특징 등을 검출하고 특징 벡터 추출

- **해석**
  - 응용에 따라 다양한 형태

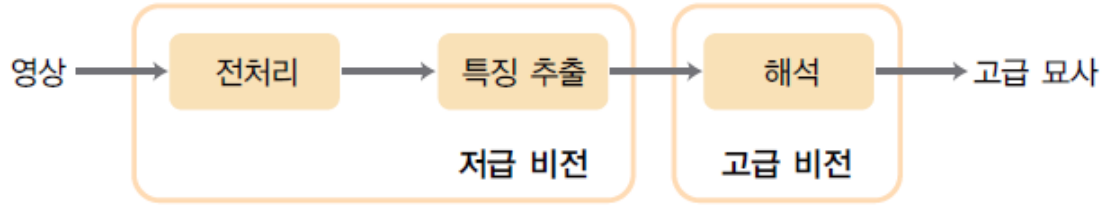
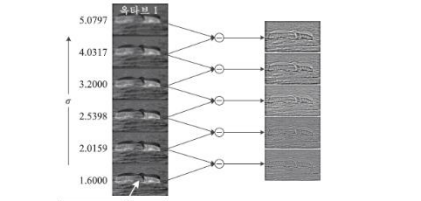


그림 1–6 컴퓨터 비전의 계층적 처리

검출

- 영상을 입력으로 받아 특징 또는 객체를 검출

- 특징 검출
  - 에지 검출
  - 지역 특징 옥타브 1 검출
- 특징 기술
  - 관심점 기술자 – SIFT, SURF
  - 영역 기술자 – 모멘트, 모양, 푸리에 기술자
  - 텍스쳐, 지역 관계 기술자 - LBP



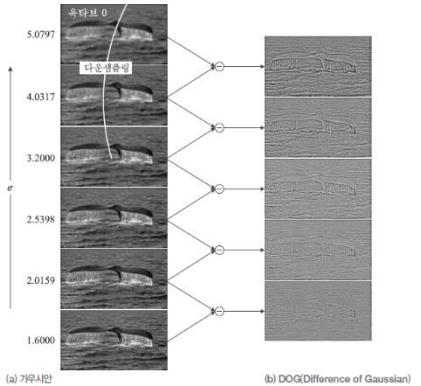


그림 4-16 SIFT가 사용하는 스케일 공간(DOG 피라미드 구조)

136	165	147	133	139	125
105	150	142	143	163	140
113	153	160	176	177	140
160	186	204	200	177	139
184	178	188	188	166	148
147	139	146	148	140	136

0	1	0
0		0
0	1	1

00110010=2+16+32=50

그림 6-17 LBP 계산

영상 분류

- 영상을 입력으로 받아 객체를 분류
- 유형별 분류
  - 문자인식
  - 딥러닝을 활용한 영상 객체 분류
- 범주 분류
- 대표적인 컴퓨터비전 대회
  - PASCAL VOC(http://host.robots.ox.ac.u
    - 분류(20),검출,분할에 대한 5가지 문제
    - 2012년 종료
  - ImageNet ILSVRC(http://www.image-ne
    - 1000종류 분류 문제
    - 2017년 인간의 성능을 능가하면서 종료
  - ICDAR RRC
    - 자연 영상에서 텍스쳐 검출

(a) 사례 인식

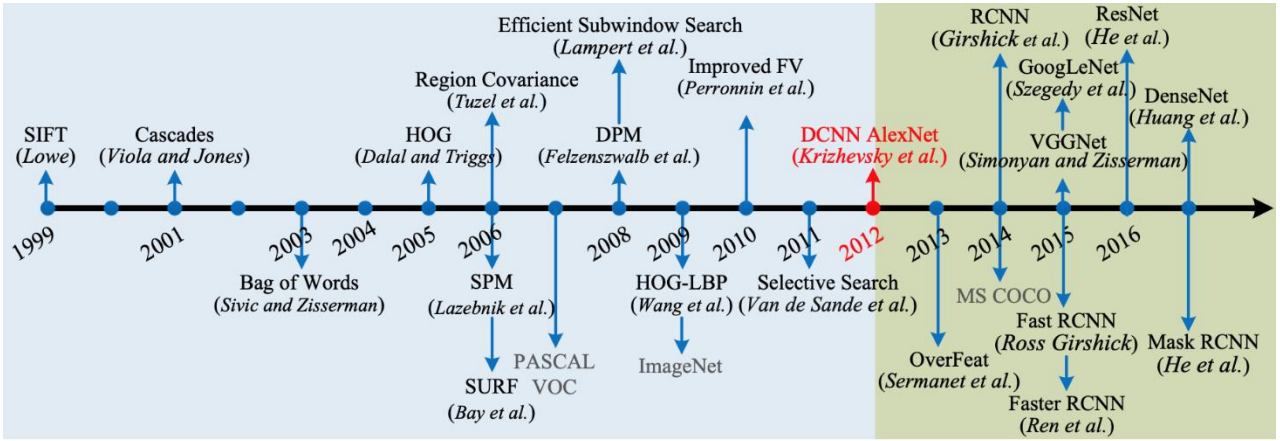
(b) 범주 인식

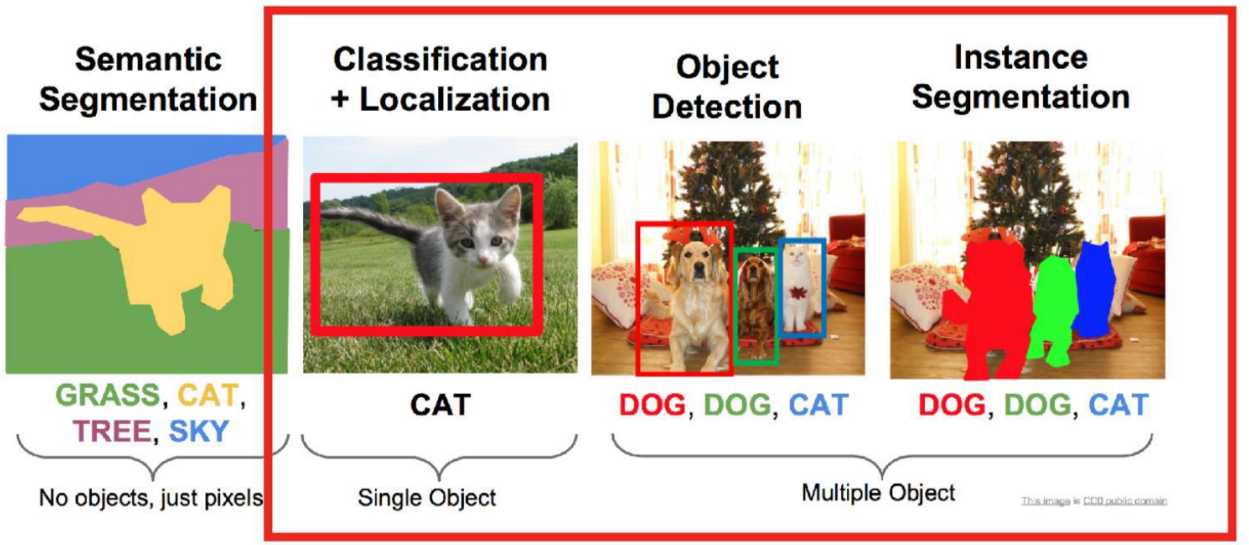
그림 9-2 사례 인식과 범주 인식

영상 인식

- 영상을 입력으로 받아 객체를 감지하고 분류하고 인식

- 객체 탐지
  - 사물 및 위치 인식
  - 얼굴인식
- 영상 분할
  - 의미론적 분할
  - 객체 분할
- 3D 객체 인식
  - Objectron





OpenCV

OpenCV 개요

- OpenCV
  - 인텔 주도로 개발된 컴퓨터 비전용 라이브러리
  - 픈 소스 소프트웨어 (BSD 라이선스)
  - 공식 웹 사이트는 http://opencv.org/이다.
- 지원 기능
  - OpenCV는 영상처리, 얼굴 인식, 물체 감지, 비디오 캡처 및 분석, 딥러닝 프레임워크인 TensorFlow, Torch/PyTorch, Caffe도 지원하고 있다.
  - 인텔의 데이터 중심 연산 최적화 라이브러리인 IPP(Integrated Performance Primitives)가 설치되어 있으면 활용
- 주요 기능
  - 영상 파일의 읽기 및 쓰기
  - 비디오 캡처 및 저장
  - 영상 처리(필터, 변환)
  - 영상이나 비디오에서 얼굴, 눈, 자동차와 같은 특정 물체를 감지
  - 비디오를 분석하여 움직임을 추정하고, 배경을 없애고, 특정 물체를 추적
  - 기계 학습 알고리즘을 사용하여 물체 인식 가능

OpenCV 개요

- **라이브러리 구성**
  - **현재 4.5 버전까지 개발**
  - **기본 기능**

패키지명	기능	패키지명	기능
core	핵심 기능	ml	기계 학습
imgproc	이미지 처리	flann	다차원 클러스터링 및 탐색
highgui	고급 GUI 및 미디어 I/O	gpu	GPU 가속
video	비디오 분석	photo	연산 사진술(photography)
calib3d	카메라 캘리브레이션 및 3D 재구성	stitching	이미지 스티칭(붙이기)
features2d	2D 특징 추출 프레임워크	gapi	그래프 API
objdetect	객체 탐지	nonfree	유료 기능
dnn	심층 신경망	contrib	기증된/실험적인 기능

  - **추가 기능 : aruco(마커 탐지), cnn_3dobj(3D 객체인식및자세추정),cudaxx(가속기능), rgbd(RGB-깊이처리), tracking(객체추적), viz(3D시각화) 등**

OpenCV 설치

- 아나콘다 환경에서 OpenCV 설치
  - conda install OpenCV 를 입력하여 설치
- 코랩 환경에서 OpenCV 설치
  - pip install opencv-pytyon 입력하여 설치
- 설치 확인
  - 파이썬 쉘(Python shell)에서 아래의 명령을 입력하여 OpenCV버전을 확인
  - >>> import cv2
  - >>> print(cv2.__version__)

```
(cm) C:₩Users₩cm>python
Python 3.6.10 |Anaconda, Inc.| (default, May  7 2020, 19:46:08) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
3.4.4
>>>
```

- 파이썬 예제 참고자료
  - https://wjddyd66.github.io/categories/#opencv

주요 함수

- 이미지 읽기 : cv2.imread()
  - 컬러 이미지를 읽으려면 cv2.IMREAD_COLOR를, 그레이스케일 모드로 읽으려면 cv2.IMREAD_GRAYSCALE을 입력
- 이미지 컬러 공간 변환 : cv2.cvtColor()
  - BGR 색공간→gray-scale로 변환 : cv2.COLOR_BGR2GRAY,
  - BGR 색공간→HSV 색공간으로 변환 : cv2.COLOR_BGR2HSV를 선택
- 이미지 사이즈 변경 : cv2.resize()
  - 이미지를 축소시킬 때는 보간법(interpolation) 설정을 cv2.INTER_AREA로, 확대시킬 때는 cv2.INTER_CUBIC 또는 cv2.INTER_LINEAR를 사용
- 이미지 보여주기 : cv2.imshow()
  - 보통 cv2.waitKey(0), cv2.destroyAllWindows()와 함께 사용
- 이미지 저장 : cv2.imwrite()
- 선/사각형/원/타원 그리기 : cv2.line(), cv2.rectangle(), cv2.circle(), cv2.ellipse()
- 텍스트 넣기 : cv2.putText()
- 이미지 채널 분리 및 병합 : cv2.split(), cv2.merge()
- 값의 범위 바꿔주기 : cv2.normalize()

실습

- OpenCV 기본 동작 실습
  - <실습:gg-07-opencv_intro>
  - 얼굴인식
    - 최근 딥러닝 기술이 아니라 OpenCV가 제공하는 패턴 인식 기술 활용
    - 사람 얼굴에는 공통된 특징이 있어서 모든 사람의 눈,코,입 부분의 명암의 매우 유사한 패턴을 가지고 있다.
    - 이러한 얼굴 고유의 특징을 데이터베이스화하여 사람의 얼굴을 이미지에서 추출하는 방법으로 캐스케이드 파일을 이용할 수 있다.
    - haarcascade 검색하면 찾을 수 있으며 아래는 Haar-like 특징 학습기를 다운로드 받는 방법이다.

필터링과 기학
학적 변환

영상의 산술 연산

- 화소 점 처리
  - 화소 점(poing)의 값을 변경하는 기술
  - 산술연산, 논리연산, 반전, 광도 보정, 히스토그램 평활화 등의 기법
- 밝기 조절 산술연산
  - 화소 값을 변환하여 밝기를 조절해 주는 가장 간단한 산술연산은 덧셈과 뺄셈
  - 이 방법은 주로 텔레비전이나 모니터의 밝기를 조절하는 데 사용
  - 영상 내의 밝기 차를 조절할 때는 주로 곱셈과 덧셈연산을 사용
- 참고자료
  - 4장 화소값처리, 디지털영상처리입문, 한빛미디어

영상의 산술 연산

- **화소 값의 덧셈 연산**
  - 화소의 밝기 값에 특정한 상수 값을 더하면 화소의 밝기 값을 증가시켜 영상을 밝게 하는 처리 기술
  - 화소 + alpha : 영상의 밝기 증가
  - 화소는 최대값인 255를 넘는 값은 모두 255로 처리
- **화소 값의 뺄셈 연산**
  - 화소의 밝기 값에 특정한 상수 값을 빼면 화소의 밝기 값을 감소시켜 영상을 밝기를 어둡게 하는 처리 기술
  - 화소 - alpha : 영상의 밝기 감소
  - 화소의 최소값이 0보다 작은 음수 값은 모두 0으로 처리
- **화소 값의 곱셈연산**
  - 화소의 밝기 값에 특정 상수 값을 곱하면 전체적으로 화소의 밝기 값이 증가해 더 밝아진다. 기존의 밝은 부분은 더욱 밝아지고, 어두운 부분은 약간 밝아진다.
  - 화소 * alpha : 영상의 밝기 차이 증가 = 뚜렷해짐
- **화소 값의 나눗셈연산**
  - 화소 값의 임의의 상수 값으로 나누면 전체적으로 화소의 밝기 값은 감소하고, 최대 밝기와 최소 밝기의 차이는 작아진다.
  - 화소 / alpha : 영상의 밝기 차이 감소 = 희미해짐

실습

- **OpenCV 산술 연산 실습**
  - **<실습:gg-08-화소_점_처리.ipynb>**

필터링

- 화소 영역 처리
  - 주변 화소값도 고려하여 공간 영역(area) 연산
  - 합성곱(convolution)으로 처리하므로 합성곱 처리라고 함
  - 엠보싱, 블러링, 샤프닝, 필터링 등 연산
- 필터링
  - 필터를 이용한 영상처리는 화소 영역 처리에 해당하며, 당연히 컨볼루션 수행
- 필터 종류
  - 유한임펄스응답(FIR, Finite Impulse Response) 필터
    - 필터의 길이가 한정
    - 필터를 설계하기가 쉽고, 이를 이용하여 신호도 쉽게 처리 가능
    - 영상처리에서 효과적인 필터링과 선형 시불변 시스템 특성 만족
  - 무한임펄스응답(IIR, Infinite Impulse Response)필터
    - 필터의 길이가 무한정
    - IIR 필터는 설계하기가 어렵고 처리도 힘들지만, 필터의 특성은 더 우수
- 참고자료
  - 6장 화소영역처리, 디지털영상처리입문, 한빛미디어
  - 12장 필터링, 디지털영상처리입문, 한빛미디어

합성곱 (Convolution)

- 임의의 디지털 신호 X[n]이 선형 시불변 시스템인 FIR 필터에 입력 되어 원하는 출력 Y[n]을 만드는 과정
- 값 세 개의 평균을 구하는 입출력 관계식

$$y[n] = \frac{1}{3}(x[n]) + x[n-1] + x[n-2])$$

- 이 과정에서 출력을 얻으려고 사용한 선형 시불변 시스템 FIR 필터

$$b[n] = \frac{1}{3}\delta[n] + \frac{1}{3}\delta[n-1] + \frac{1}{3}[n-2]$$

$$\delta[n] = \begin{cases} 1, n = 0 \\ 0, n \neq 0 \end{cases}$$

영상의 공간 필터링

- 필터링을 이용한 영상처리는 2차원 합성곱을 수행하게 됨
  - 영상의 공간 필터링은 다음과 같이 크기가 M*N인 FIR 필터 마스크 h[x,y]와 크기가 M*N인 영상 간의 2차원 합성곱을 수행하는 것

$$y[x,y] = \sum_{k=0}^{M} \sum_{l=0}^{N} b[k,l]x[x-k, y-l]$$

  - 사용되는 필터 마스크를 컨벌루션 마스크 또는 회선 마스크
- 공간 필터링(Spatial Filtering)
  - 영상에 있는 공간 주파수 대역을 제거하거나 강조하는 필터 처리
  - 공간 주파수란 단위 공간에서 같은 화소 값이나 같은 색이 반복되는 횟수
  - 고주파 : 밝기 변화가 빠르거나 색의 변화가 급격한 곳
  - 저주파 : 밝기 변화가 늦거나 색의 변화가 적은 곳

a	b	c
d	e	f
g	h	i

저주파 통과 필터링 (LPF)

- 신호 성분 중 저주파 성분은 통과시키고 고주파 성분은 차단하는 필터
- 잡음을 제거하거나 흐릿한 영상을 얻을 때 주로 사용
- 고주파 성분을 제거하므로 고주파 차단 필터

- OpenCV를 이용해 다양한 저주파 필터(Low-pass filter)를 이용해 이미지를 부드럽게, 즉 블러링(Blurring) 처리를 하는 내용이다.
- 2D Convolution, 같은 의미로써 이미지 필터링(Filtering)은 고주파, 또는 저주파 필터를 이용해 이미지를 처리하는 것을 의미합니다. 여기서 필터는 예를들어 아래와 같은 행렬이

$$K = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

고주파 통과 필터링(HPF)

- 신호 성분 중 고주파 성분은 통과시키고 저주파 성분은 차단하는 필터
- 저주파 성분을 차단하므로 저주파 차단 필터
- 고주파 통과 필터링은 샤프닝과 같은 처리 방법
- 흐려진 영상을 개선하여 첨예화 하는 결과 영상을 생성
- 샤프닝 필터
  - 고주파 통과 필터에서 발생하는 낮은 공간 주파수의 성분이 손실되는 문제점을 보완해 주는 회선 마스크
  - 샤프닝 필터링된 영상은 원본 영상에 고주파 통과 필터링된 영상을 합한 것과 비슷한 결과
  - 10장의 엣지검출 때 구현

영상의 잡음 제거와 프로파일 분석

- 잡음 제거 방법
  - 형태학적 처리의 침식연산은 일반적인 필터링 처리와는 달리 잡음 제거 가능
  - 대표적인 비선형 공간 필터링으로 잡음을 제거하는 방법인 중간값 필터링과 최소/최대 필터링

중간 값 필터링 잡음 제거

- 중간 값 필터
  - 이웃 화소의 값을 오름차순으로 정렬한 뒤 가운데 값을 출력으로 선택
  - 영상에 스파크처럼 급격한 색 변화가 있는 임펄스 잡음을 제거용 사용
  - 기존의 평균 필터를 이용한 선형 공간 필터링 방법에 비해 블러링 현상이 적고 객체의 경계를 잘 보존한다는 장점.
  - 즉, 평균 필터를 이용한 방법의 단점을 보완한 방법이다.
  - 중간 값을 구하려고 비교하는 과정에서 많은 시간이 소모된다는 단점
- 중간 값 필터링으로 잡음제거
  - 중간 값 필터는 경계 부분을 잘 보존하는 편이지만, 좀더 세부적인 경계 영역까지도 보존할 수 있는 방법 필요
  - 가중 중간 값 필터(Weighted Median Filter)가 제안
    - 가중치를 설정하여 영상 내의 세부 정보인 경계 영역을 보존하면서 동시에 잡음을 제거하는 특성

최대/최소 값 필터링 잡음 제거

- 최대/최소 필터링
  - 중심 화소를 이웃 화소의 중간 값으로 치환하는 대신 최소값이나 최대값으로 치환하는 방법
  - 중간 값 필터링과 비슷한 방법
  - 영상에 있는 극단적인 임펄스 값을 제거하는 데 사용되는 필터링 기법으로, 의료 영상에 주로 사용
  - 그러나 혼합된 임펄스 잡음을 제거하기는 어렵다.
- 최소값 필터링
  - 정렬된 값 중에서 최소값을 선택하여 밝은 임펄스 값을 제거
  - 출력 영상의 전체 밝기가 감소
- 최대값 필터링
  - 반대로 정렬된 값 중에서 최대값을 선택하여 어두운 임펄스 값을 제거
  - 출력 영상의 전체 밝기는 증가

실습

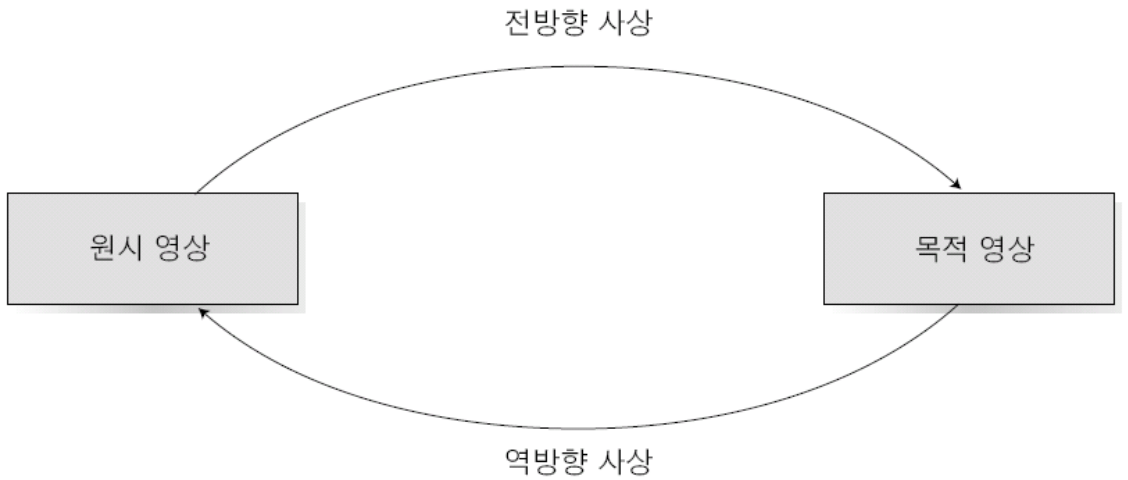
- OpenCV 저주파 통과 필터링 실습
  - <실습:gg-08-저주파_통과_필터링.ipynb>
  - OpenCV에서 제공하는 4가지 필터 정리
    - 평균 필터
    - 가우시안 필터링(Gaussian Filtering)
    - Median Filtering - 잡음 제거 효과가 뛰어남
    - Bilateral Filtering - 잡음 제거 효과가 좋으면서, 가장자리를 보존하는데 효과적. 단, 속도가 느리다는 단점

기하학적 변환

- 기학학적 변환
  - 영상을 구성하는 화소의 공간적 위치를 재배치하는 과정이다.
  - 재배치되는 화소가 어떤 것이냐에 따라
    - 입력 영상을 출력 영상으로 화소의 위치를 변환하는 과정으로 전방향 사상
    - 출력 영상을 입력 영상으로 화소의 위치를 변환하는 과정으로 역방향 사상
  - 기본 형태에 따라
    - 선형 기하학적 변환 : 직선 처리처럼 선형적으로 처리하는 방법으로, 평행이동, 회전, 스케일링(Scaling)등 화소의 재배치를 수행
    - 비선형 기하학적 변환 : 영상을 찌그러뜨리고 구부려서 곡선으로 처리하는 방법으로, 위핑과 모핑
- 참고자료
  - 8장 기하학적변환, 디지털영상처리입문, 한빛미디어

기하학적 변환의 사상

- 사상(Mapping)
  - 주어진 조건에서 현재의 데이터를 원하는 목표로 만드는 것
    - 원시 영상의 화소가 목적 영상의 화소 위치로 이동하면, 원시 영상 화소가 목적 영상의 화소로 대응
  - 사상을 전방향 사상과 역방향 사상으로 분류
  - 디지털 영상처리에서는 역방향 사상을 새로운 화소 값을 설정하는 데 아주 유용하게 사용



[그림 8-8] 전방향과 역방향 사상의 개념

전방향 사상

- 영상처리에서 전방향 사상
  - 입력 영상의 모든 화소에서 출력 영상의 새로운 화소 위치를 계산하고, 입력 화소의 밝기 값을 출력 영상의 새로운 위치에 복사하는 방법
- 문제점
  - 서로 다른 입력 화소 두 개가 같은 출력 화소에 사상되는 것이다. 이것을 오버랩(Overlap) 문제
  - 입력 영상에서 임의의 화소가 목적 영상의 화소에 사상되지 않을 때다. 이를 홀(Hole)문제

역방향 사상

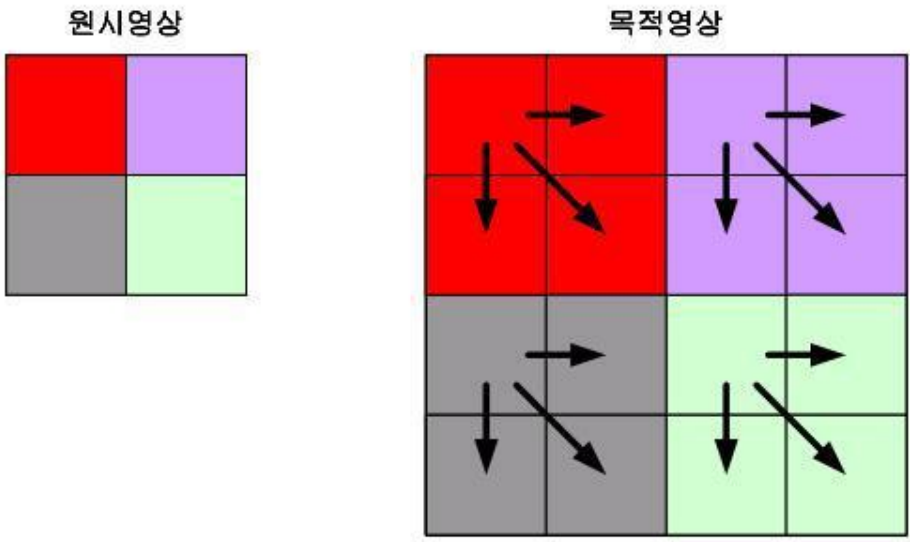
- 역방향 사상
  - 전방향 사상과는 반대되는 개념
  - 목적 영상에서 원시 영상의 화소 값을 찾는 것
  - 목적 영상의 화소를 조사하여 몇 가지 역변환으로 원시 영상의 화소를 구한 뒤 목적 영상의 화소 값을 생성하려고 사용

보간법

- 보간법
  - 기하학적 처리 결과가 완전한 사상이 되지 못하면 목적 영상의 일부 화소가 값을 할당받지 못하는 상황이 발생하면 화소 값을 할당받지 못한 목적 영상의 품질은 아주 좋지 못함
  - 빈 화소에 값을 할당하여 좋은 품질의 영상을 만드는 방법
  - 화소의 값을 할당받지 못한 채 목적 영상을 만드는 대표적인 기하학적 처리가 바로 영상의 확대
  - 복잡한 알고리즘의 보간법은 영상의 질을 향상시키나 처리 시간(Running Time)이 많이 드는 단점이 있으므로 적용 대상에 맞게 적절한 보간 함수를 선택
- 대표적인 보간법
  - 가장 인접한 이웃 화소 보간법(Nearest Neighbor Interpolation)
  - 양선형 보간법(Bilinear Interpolation)
  - 3차 회선 보간법(Cubic Convolution Interpolation)
  - B-스플라인 보간법(B-Spline Interpolation)

가장 인접한 이웃 화소 보간법

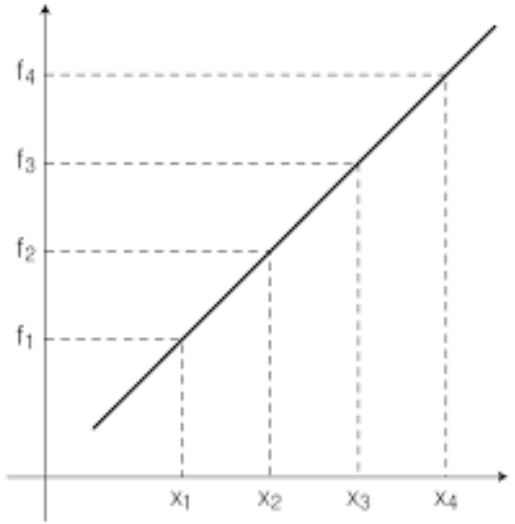
- 값을 할당받지 못한 목적 영상의 화소에서 가장 가깝게 이웃한 원시 화소의 값을 할당받은 목적 영상의 화소 값을 복사해서 사용하는 것
- 가장 인접한 이웃 화소 보간법은 단순히 이웃 화소를 복사하여 사용하므로 처리 속도가 빠르다는 장점
- 그러나 새로운 화소 값을 계산하지 않고 입력 화소 내에서만 찾기 때문에 원래의 영상과 전혀 다른 영상을 출력하는 오류가 발생



[그림 8-14] 가장 인접한 이웃 화소 보간법의 동작

선형 보간법

- **선형 보간법(Linear Interpolation)**
  - 원시 영상의 화소 값 두 개를 이용하여 원하는 좌표에서 새로운 화소 값을 계산하는 간단한 방법
  - 선형 보간을 수행하는 과정으로, 두 화소의 좌표(x1, f1)과 (x2, f2)를 이용하여 두 화소 사이에 위치하는 새로운 화소 f3을 보간한다.

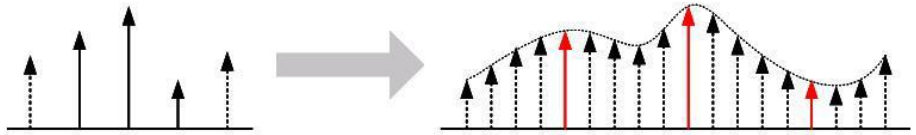


  - f3은 다음 식으로 얻을 수 있다

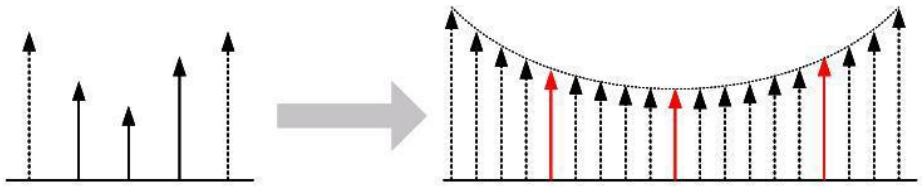
$$f3 = \frac{(x_3 - x_1)}{(x_2 - x_1)}(f_2 - f_1) + f_1$$

고차 보간법

- 고차 보간법
  - 더 많은 이웃 화소를 참조하므로 값을 할당받지 못한 화소 값을 쉽게 추정 가능
- 대표적인 방법으로 3차 회선과 B-스플라인
- 3차원 회선 보간법
  - 4 X 4의 이웃 화소를 참조하여 보간을 수행
  - 양선형 보간법보다 더 많은 화소 참조하므로 보간 영상 품질도 더 좋다
  - 이웃 화소를 16개 참조하므로 계산 시간이 더 소요된다.



- B-스플라인 보간법
  - 이상적인 보간 함수는 저주파 통과 필터인데, B-스플라인 함수는 그중에서도 상당히 좋은 저주파 통과 필터
  - 따라서 B-스플라인 함수는 보간 함수 중에서 가장 스무딩한 영상 출력



실습

- OpenCV 기하학적 변환 실습
  - <실습:gg-08-기하학적_변환.ipynb>
  - OpenCV에서 제공하는 기하학적 변환
    - 스케일링
    - 회전
    - Perspective Transform

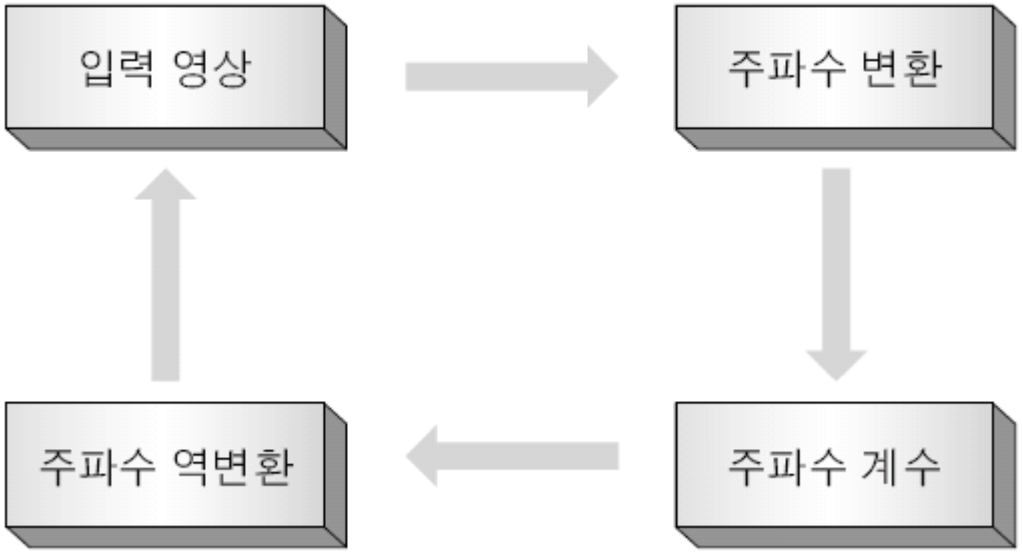
영상 신호처리

영상 신호처리 개요

- 영상 변환
  - 영상의 개선, 복원, 압축, 해석 등 다양한 영상처리 작업을 공간에서 처리하는 것보다 더 쉽고 효율적으로 수행할 수 있게 함
  - 대부분 공간 영역의 영상을 주파수 영역으로 변환하는 방식을 사용
  - 주파수 : 영상에서 화소 밝기의 변화 정도를 나타내는 것은 화소 값의 변화율
  - 주파수는 밝기가 얼마나 빨리 변화하느냐에 따라서 고주파와 저주파로 분류
- 영상을 공간 주파수 영역으로 변환하면 저주파와 고주파 성분으로 분리됨
  - 영상의 일정 주파수 영역을 조작하여 원래의 영상을 개선하는 작업 가능
  - 높은 주파수 성분을 낮추면 섬세한 부분이 사라지고, 부드럽고 엉성한 영상으로 변한다.
  - 반대로 낮은 주파수 성분을 낮추면 엉성한 부분이 사라지면서 섬세한 부분에 해당하는 경계가 강조
- 참고자료
  - 13장 영상변환, 디지털영상처리입문, 한빛미디어

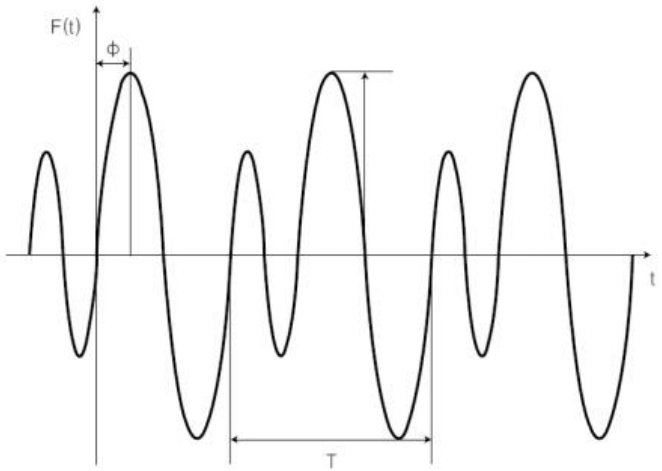
영상 주파수 변환

- 주파수 변환
  - 모든 영상 데이터는 주파수 성분으로 변환이 가능
  - 공간 영역 형태의 영상을 주파수 영역 형태의 기본 주파수로 분리하는 것
  - 정규적인 변환이 성립하려면 역변환도 성립되어야 함
  - 주파수 형태의 영상을 공간 형식으로 변환하는 역주파수 변환이 반드시 있어야 한다



푸리에 변환

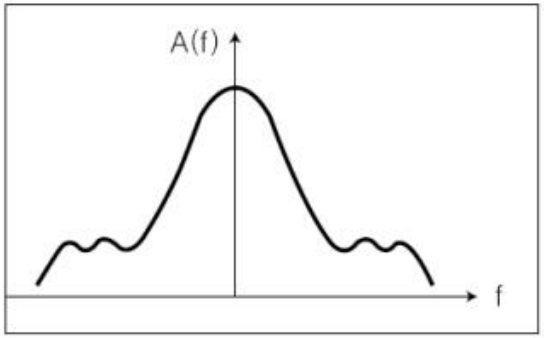
- 푸리에 변환(Fourier Transform)
  - 주파수 영역으로 변환하는 가장 일반적인 방법
  - 푸리에는 열 전파와 확산을 연구하여, 주기성이 있는 신호는 연속된 정현파의 조합으로 표현할 수 있다는 결과를 발표
- 신호를 구성하는 세 가지 요소
  - 주기 T : 반복되는 시간
  - 진폭 A : 파형의 크기.0에서 양의 최대 높이까지의 거리
  - 위상 ϕ : 파형의 시작이 얼만큼 지연되고 선행되었는지를 나타내는 시간 차이



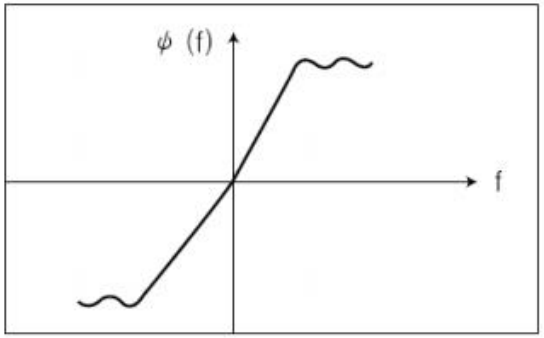
푸리에 변환

(a) 시간 영역에서의 신호 파형

푸리에 변환



(b) 주파수 영역에서의 신호 크기



(c) 주파수 영역에서의 신호 위상

푸리에 변환

- **연속 푸리에 변환**
  - 연속적인 시간 영역의 신호를 주파수 영역으로 변환하는 것

$$\Im\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-j2\pi ft}\,dt$$

  - 2차원 연속 푸리에 변환 공식

$$\Im^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f)e^{j2\pi ft}\,df$$

$$\Im\{g(x,y)\} = G(f_x, f_y) = \iint_{-\infty} g(x,y)e^{-j2\pi(f_x x + f_y y)}\,dxdy$$

$$\Im^{-1}\{G(f_x, f_y)\} = g(x,y) = \iint_{-\infty} G(f_x, f_y)e^{j2\pi(f_x x + f_y y)}\,df_x df_y$$

이산 푸리에 변환

- **이산 푸리에 변환**
  - 디지털 영상은 아날로그 신호가 아니고 디지털 데이터이므로, 연속 푸리에 변환에 직접적으로 적용할 수 없음.
  - 이산 푸리에 변환(Discrete Fourier Transformation)은 디지털 신호를 주파수 영역으로 변환해 줌.
  - 이산 푸리에 변환 공식: 연속 푸리에 변환의 적분을 합(Sum)으로 변경

$$G(\hat{f}) = \frac{1}{N}\sum_{n=0}^{N-1} g[n] e^{-j2\pi \hat{f}\frac{n}{N}}$$

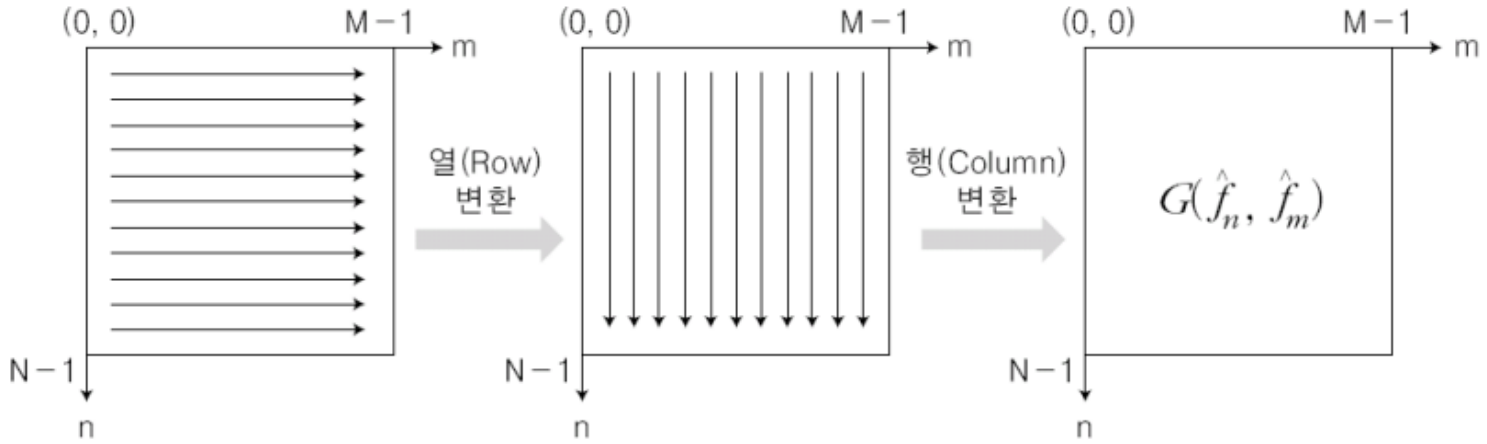
$$g[n] = \sum_{\hat{f}=0}^{N-1} G(\hat{f}) e^{j2\pi \hat{f}\frac{n}{N}}$$

$$G(\hat{f}_n, \hat{f}_m) = \frac{1}{MN}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} g[n,m] e^{-j2\pi(\hat{f}_n n/N + \hat{f}_m m/M)}$$

$$g[n,m] = \frac{1}{MN}\sum_{\hat{f}_n=0}^{M-1}\sum_{\hat{f}_m=0}^{N-1} G(\hat{f}_n, \hat{f}_m) e^{j2\pi(\hat{f}_n n/N + \hat{f}_m m/M)}$$

고속 푸리에 변환(Fast Fourier Transform: FFT)

- 이산 푸리에 변환은 복잡하고 연산량이 많아 하드웨어를 구현할 때 처리 속도가 늦어진다는 단점
- 고속 푸리에 변환(FFT, Fast Fourier Transform)은 이산 푸리에 변환 공식에서 반복 계산을 제거하면 변환을 빠르게 수행할 수 있다.
- 분리성 때문에 1차원 DFT는 효과적으로 1차원 FFT를 적용할 수 있어 계산량을 크게 줄일 수 있고, 하드웨어 구현이 용이해진다. 결국, 2차원 FFT가 수행되는 것



순방향 고속 푸리에 변환

- FFT를 영상에 적용하려면 필수적으로 영상의 크기도 2의 지수 승이여야 한다.
- 영상의 크기가 2의 지수 승이 아니라면 0의 값을 삽입하여 강제적으로 2의 지수 승을 만들고, 1차원 FFT는 두 단계로 구현 된다.
  - 1) 스크램블링 – 재귀적인 DFT 계산 주기와 맞추려고 데이터를 적절히 재배치

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk}\ ,\ k=0,1\cdots,N-1$$

  - 2) 버터플라이 함수 적용 – 데이터를 점(Pointer)의 집합으로 나눠 이웃한 점의 DFT 변환 수행

$$W_N^{nk}=e^{-j2\pi(nk/N)}$$

  - 3) 결국, 다음과 같이 정리할 수 있다.

$$X(k)=G(k)+W_N^kH(k)$$

이산 코사인 변환(Discrete Cosine Transform: DCT)

- 영상을 압축하는 가장 효과적인 방법임을
- 이산 코사인 변환은 푸리에 변환의 실수 부분의 코사인과 매우 비슷
- 기저 함수가 코사인 함수가 된다.
- 실수부만 다루므로 신호 처리를 효과적으로 수행
- 1차원 이산 코사인 변환

$$F(u) = k(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right], \qquad u = 0,1,\ldots,N-1$$

$$f(x) = \sum_{x=0}^{N-1} k(u)F(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right], \qquad x = 0,1,\ldots,N-1$$

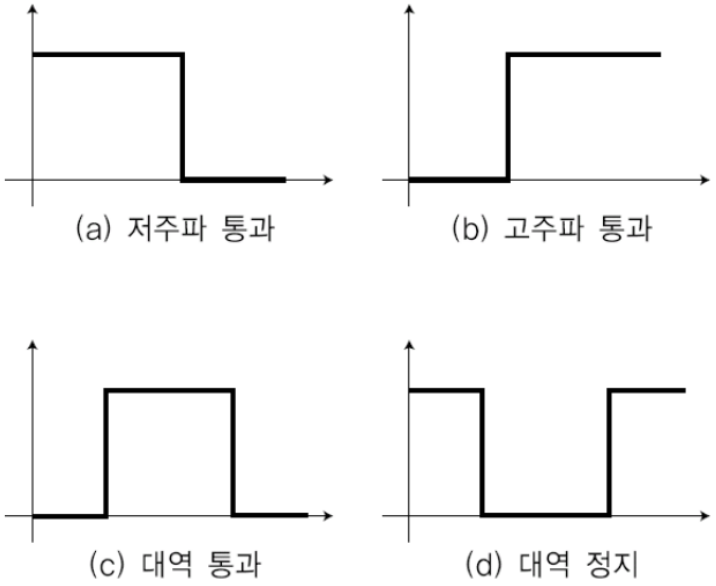
- 2차원 이산 코사인 변환

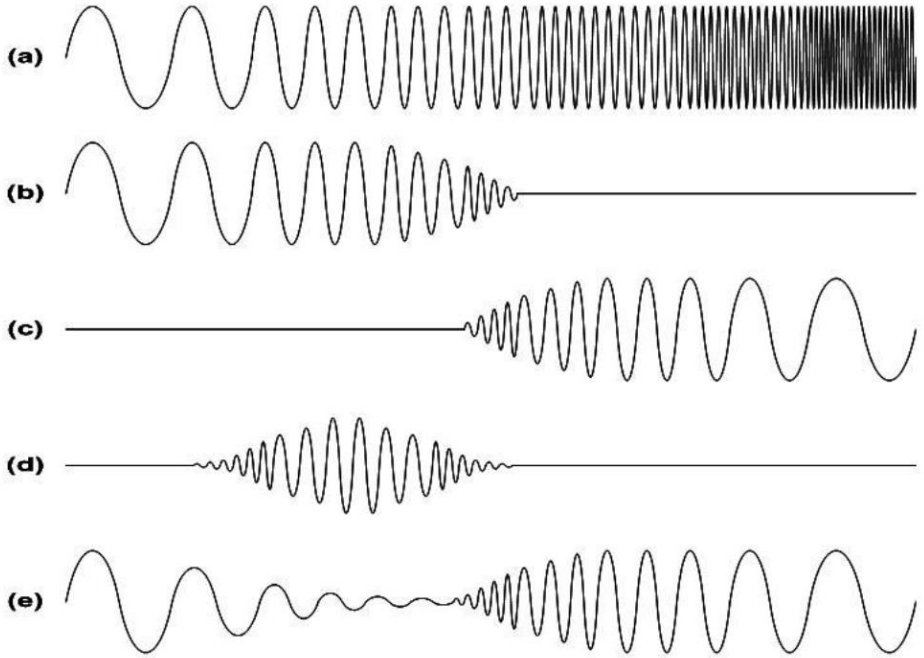
$$F(u,v) = k(u)k(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} k(u)k(v)F(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

주파수 영역 필터링

- 영상의 푸리에 변환을 수행하는 목적 중 하나는 주파수 영역에서 필터링을 수행하기 위해서
- 푸리에 변환 뒤 주파수 영역에서의 필터링은 영상에 포함된 주파수 성분을 파악하여 이를 토대로 영상에 포함된 주파수 성분을 필터링하는 것
- 기본적인 필터의 종류





주파수 영역 필터링 수행 방법

- 주파수 영역에서 필터링의 수행은 공간 영역에서의 필터링보다 쉽다
- 공간 영역에서의 필터링은 컨벌루션으로 수행하지만, 주파수 영역에서는 영상의 주파수 성분과 필터의 주파수 성분을 곱해서 해결한다. 이것을 컨벌루션 정리(Convolution Teorem)라고 한다.
- 다음은 이 정리를 수식으로 표현한 것이다. 컨벌루션 연산의 기호를 *로 표시

$$f(x, y)*h(x, y) \Leftrightarrow F(u, v)\times H(u, v)$$
$$F(u, v)*H(u, v) \Leftrightarrow f(x, y)\times h(x, y)$$

  - 여기서 f(x,y)와 h(x,y)는 공간 영역에서의 영상 데이터와 필터 계수를 나타내고, F(u,v)와 H(u,v)는 주파수 영역에서 영상 주파수 데이터와 필터 계수의 주파수 데이터다
  - u와 v는 x와 y방향의 주파수 성분을 나타낸다.
- 필터에 주파수 영역의 마스크가 주어진다면 주파수 영역에서 필터링은 다음 순서를 따른다
  - (1) 영상의 푸리에 변환을 구한다.
  - (2) 푸리에 변환된 영상과 필터 마스크를 곱한다
  - (3) (2)의 결과에 역푸리에 변환을 구한

주파수 영역 저주파 및 고주파 필터링

- 주파수 영역에서 필터링을 수행하려면, 먼저 주파수 영역의 필터 마스크를 만들어야 한다.
- 이상적인 저주파 통과 필터 생성
  - 고주파 성분을 감쇄시켜 영상을 흐릿하게 만드는 이상적인 저주파 통과 필터는 원점에서 어느 거리 내의 저주파 성분은 1을 곱해 통과시키고, 거리 밖의 고주파 성분은 0을 곱해 차단하도록 주파수 영역을 설계
  - 이상적인 저주파 통과 필터 수식

$$H(u,v) = \begin{cases} 1, & r(u,v) \le r_0 \\ 0, & r(u,v) > r_0 \end{cases}$$

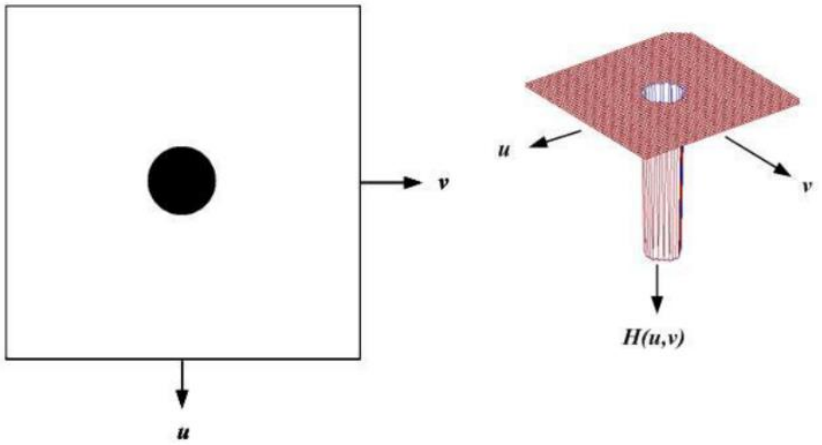
  - r_0 : 필터의 반경이며, 차단 주파수

이상적인 고주파 통과 필터 생성

- 저주파 성분을 감쇄시켜 영상을 선명하게 만드는 이상적인 고주파 필터의 생성은 저주파 통과 필터에서 다음 수식으로 얻을 수 있다

$$H_{high}(u,v) = 1 - H_{low}(u,v)$$

- 이것은 원점에서 어느 거리 밖의 고주파 성분은 1로 곱해서 통과시키고, 거리 내의 저주파 성분은 0으로 곱해서 차단하게 된다. 다음은 이상적인 고주파 통과 필터의 특성을 나타낸다.



  - (a)는 주파수 영역에서의 필터마스크, (b)는 3차원 필터의 필터 응답

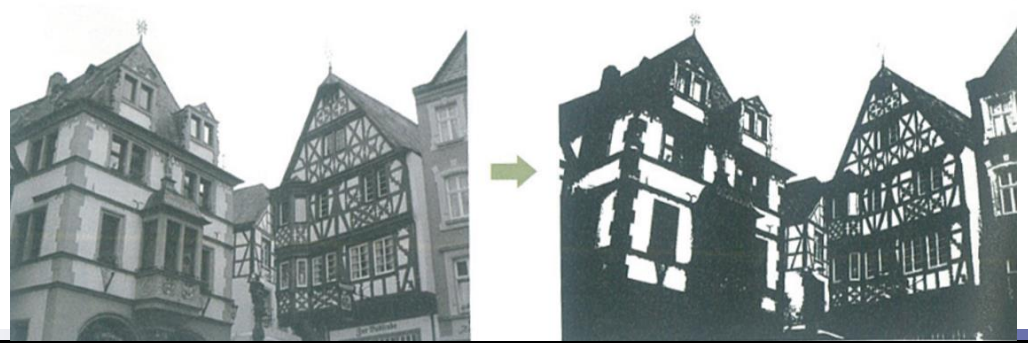
실습

- OpenCV 영상변환 실습
  - <실습:gg-09-영상_변환.ipynb>
  - OpenCV에서 제공하는 영상 변환
    - DFT
    - DCT
    - 필터링
    - 복원

영상 분할

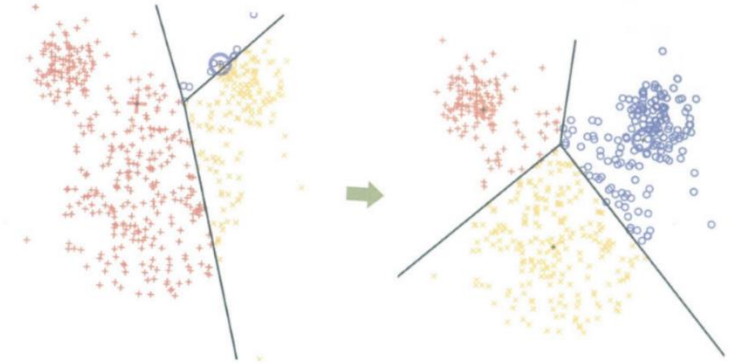
영상 분할 개요

- 영상 분할(image segmentation)
  - 영상 안의 화소를 의미 있는 영역으로 분할
  - 즉, 영상의 모든 화소에 어떤 레이블을 붙이는 것
- 영상 분할 방법
  - 에지 기반 방법 : 에지를 사용하여 물체의 윤곽선을 추적
  - 클러스터링(Clustering) – 영역 성장법을 사용
  - 히스토그램이나 무늬를 이용하는 분할방법
  - 최근에는 딥러닝을 이용하여 영상을 분할하는 방법도 많이 사용
  - 이진화
    - 가장 간단한 영상 분할 방법
    - 어떤 임계값을 정하고 이 값을 기준으로 흑백 영상을 만듦



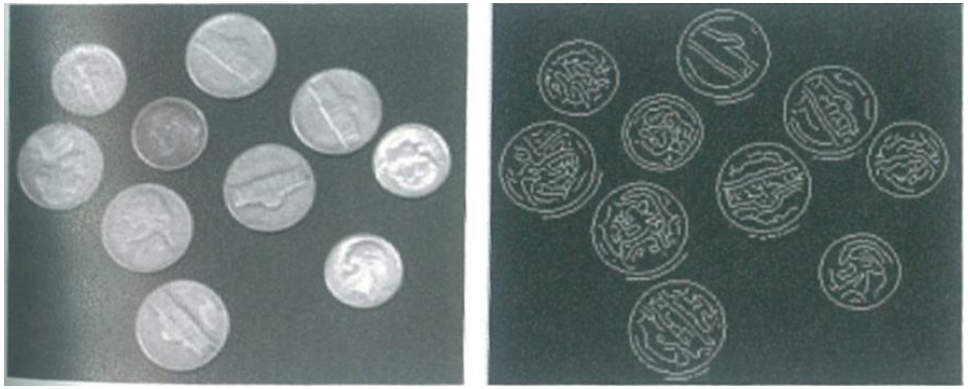
K-Means

- K-means 알고리즘
  - 영상을 K개의 클러스터로 나누는 반복적인 알고리즘
  - 알고리즘의 핵심 절차
    - 1. 임의로 K개의 클러스터의 중심을 선택한다.
    - 2. 각 화소를 화소와 클러스터 중심 사이의 거리가 최소가 되는 클러스터에 할당한다.
    - 3. 클러스터의 모든 화소를 평균하여 클러스터 중심을 다시 계산한다.
    - 4. 클러스터링이 수렴할 때까지 2단계와 3단계를 반복한다
  - 거리는 화소와 클러스터 중심 간의 제곱값이나 절대 차이값으로 계산
    - 화소 간의 거리는 일반적으로 화소의 색상, 밝기, 질감, 위치 또는 이러한 요소의 가중치 합을 기반
  - K값은 수동으로 또는 무작위로 결정된다.



에지 검출

- 에지 검출
  - 영상 처리에서 가장 잘 발달된 분야
  - 영역 경계와 에지는 밀접한 관련
    - 영역 경계에서 화소의 밝기가 급격하게 변경되기 때문
    - 에지 검출 기술은 영상 분할의 기초 자료로 사용되어 왔다.
  - 문제는 에지 검출로 식별된 에지는 종종 연결이 끊어진다는 점
  - 영상에서 물체를 분할하면 영역 경계가 닫혀 있어야 한다. 따라서 끊어진 에지들을 연결시키는 방법이 필요하다.



이진화

- 이진화는 가장 간단한 영상 분할 방법
- 이진화는 기준이 되는 값을 정하고 이 값보다 낮은 화소는 전부 0으로 만들고 이 값보다 큰 화소들은 전부 1로 만드는 처리
- 여기서 기준이 되는 값을 임계값
- 이진화를 사용하면 영상에서 전경과 배경을 분리
- 이진화의 수학적인 표현

$$dst(x,y) \simeq \begin{cases} 1\,, src(x,y) > T\text{인 경우} \\ 0\,, src(x,y) \leq T\text{인 경우} \end{cases}$$

  - src(x,y) 는 그레이스케일 영상
  - dst(x,y)=0 인 화소의 집합을 배경영역
- OpenCV에서 이진화를 지원하는 함수는 threshold()
  - threshold(src, dst, thresh, maxval, type)

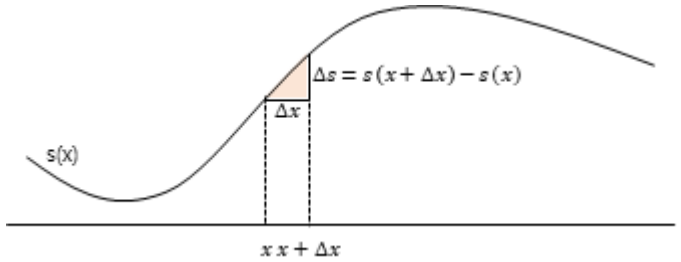
실습

- OpenCV 이진화 실습
  - <실습:gg-09-이진화.ipynb>
  - OpenCV에서 제공하는 threshold 함수
    - THRESH_BINARY는 임계값을 넘는 픽셀은 maxVal로 변경한다. 그렇지 않으면 00이 되며, THRESH_BINARY_INV는 THRESH_BINARY의 반대 작업을 한다. 즉, 임계값을 넘는 픽셀은 00이 되고 그렇지 않으면 maxVal0이 된다. 즉 물체 영역은 0, 배경은 255로 만들 수 있다.
    - THRESH_TRUNC는 화소들의 최대값을 thresh로 만든다. 즉 픽셀이 thresh보다 크면 thresh로 자르고 그렇지 않으면 화소의 값은 변경되지 않는다.
    - THRESH_TOZERO는 픽셀이 thresh보다 크면 값이 유지된다. 그렇지 않으면 0으로 된다.
    - THRESH_TOZERO_INV는 픽셀이 thresh보다 크면 00이 된다. 그렇지 않으면 값을 유지

미분 연산자

- 수학에서 변화를 측정하는 기초 이론은 미분
- 다음 그림과 식은 연속함수 s 를 미분하여 도함수를 구하는 원리

$$s'(x) = \frac{ds}{dx} = \lim_{s \to 0} \frac{s(x+\triangle x) - s(x)}{\triangle x}$$



- 컴퓨터 비전이 다루는 디지털 영상은 연속 공간이 아니라 이산 공간
  - 이산 공간에서 도함수를 근사화하는 방법 – 에지 연산자 | -1 | 1 |

$$f'(x) = \frac{fx}{dx} = \frac{f(x+\triangle x) - f(x)}{\triangle x} = f(x+1) - f(x)$$

  - 예)

0	1	2	3	4	5	6	7	8	9
2	2	3	2	3	5	9	9	8	9
0	1	−1	1	2	4	0	−1	1	−
0	0	0	0	0	1	0	0	0	−

디지털 영상 f
f의 도함수 f'
|f' |의 이진화

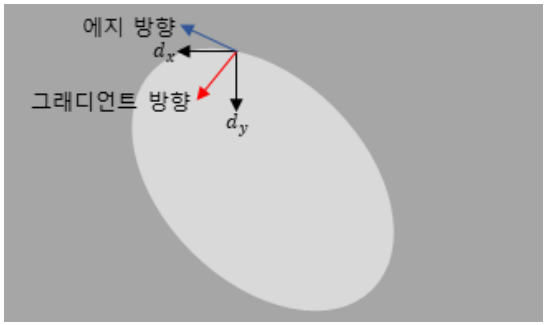
에지 검출

- 그래디언트
  - 벡터이므로 에지 강도와 에지 방향을 구할 수 있다.

그래디언트 $\nabla f = (\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}) = (d_y, d_x)$

에지 강도 $S(y,x) = magnitude(\nabla f) = \sqrt{d_y^2 + d_x^2}$

그래디언트 방향 $D(y,x) = actan(\frac{d_y}{d_x})$



  - 에지 방향은 그래디언트 방향에 수직이다. 위의 그림에 표시된 경계선 상의 한 점을 보면 이 점에 마스크 m_x 를 적용하면 음수가 되어 dx 는 왼쪽을 가리키고, dy는 양수가 되어 아래쪽을 가리킴
  - 그래디언트 방향이 정해지고, 그에 수직을 이루도록 에지 방향이 결정
  - 에지 방향은 [0,360°]범위를 갖는데, 이 범위는 보통 8-방향으로 양자화
- 참고자료
  - 7장 에지검출, 디지털영상처리입문, 한빛미디어

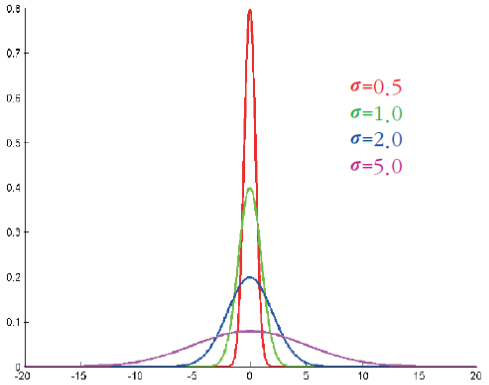
실습

- **OpenCV 에지 검출 실습**
  - **<실습:gg-10-소벨_마스크를_이용한_에지_검출.ipynb>**

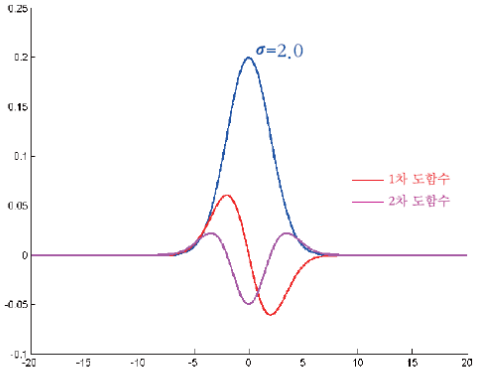
라플라시안

- 1980년 Marr Hilderth가 발표한 논문은 에지 검출의 새로운 계기
- 여기서는 1차 미분 대신 2차 미분을 사용
  - 미분을 적용하기 전 가우시안으로 스무딩하는 전처리 과정을 중요시
  - 미분은 잡음을 증폭하므로 스무딩은 매우 중요하다. 특히 2차 미분은 미분을 두 번 수행하므로 잡음 증폭이 더욱 심하다.
  - 가우시안을 사용하는 두 번째 이유는 가우시안의 매개변수 σ을 조절하여 다중 스케일 효과를 얻을 수 있기 때문
  - σ 조절해 스무딩 정도 및 에지의 스케일을 조절
    - σ 를 크게 하면 영상의 디테일이 사라져 큰 물체의 에지만 추출
    - 반대로 작게 하면 물체의 디테일에 해당하는 에지까지 추출
  - 1차원 가우시안 함수

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



(a) σ에 따른 가우시안 함수



(b) 가우시안 함수의 미분

그림 3-11 가우시안 커널

라플라시안

- 라플라시안 $\nabla^2 f$
  - 2차 편도 함수

$$\nabla^2 f(y,x) = \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial x^2}$$

  - 이산 공간의 식

$$\begin{aligned}\nabla^2 f(y,x) &= \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial x^2} \\ &= (f(y+1,x) + f(y-1,x) - 2f(y,x)) + (f(y,x+1) + f(y,x-1) - 2f(y,x)) \\ &= f(y+1,x) + f(y-1,x) + f(y,x+1) + f(y,x-1) - 4f(y,x)\end{aligned}$$

  - 필터

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

LOG 필터의 설계

- **LOG(Laplacian of Gaussian)필터**
  - 가우시안을 이산 필터로 적용한 후, 라플라시안을 이산 필터로 근사화하여 적용
  - 즉, 두 번에 걸쳐 오차를 감수하는 근사화를 수행
  - 컨볼루션도 두 번 수행하기 때문에 계산 효율도 낮음

$$LOG(y,x) = \nabla^2(G(y,x) \oplus f(y,x)) = (\nabla^2 G(y,x)) \oplus f(y,x)$$

$$\begin{aligned}
\nabla^2 G(y,x) &= \frac{\partial^2 G(y,x)}{\partial y^2} + \frac{\partial^2 G(y,x)}{\partial x^2} = \frac{\partial}{\partial y}\left(\frac{\partial G(y,x)}{\partial y}\right) + \frac{\partial}{\partial x}\left(\frac{\partial G(y,x)}{\partial x}\right) \\
&= \frac{\partial}{\partial y}\left(-\left(\frac{y}{\sigma^2}\right)\frac{1}{2\pi\sigma^2}e^{-\frac{y^2+x^2}{2\sigma^2}}\right) + \frac{\partial}{\partial x}\left(-\left(\frac{x}{\sigma^2}\right)\frac{1}{2\pi\sigma^2}e^{-\frac{y^2+x^2}{2\sigma^2}}\right) \\
&= \left(\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right)\frac{1}{2\pi\sigma^2}e^{-\frac{y^2+x^2}{2\sigma^2}} + \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right)\frac{1}{2\pi\sigma^2}e^{-\frac{y^2+x^2}{2\sigma^2}} \\
&= (y^2 + x^2
\end{aligned}$$

$$\nabla^2 G(y,x) = \left(\frac{y^2 + x^2 - 2\sigma^2}{\sigma^4}\right) G(y,x)$$

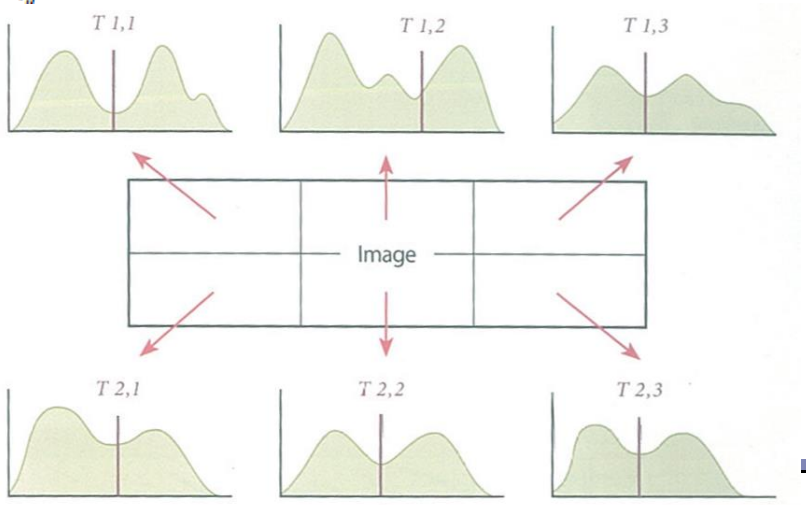
실습

- **OpenCV 라플라시안 필터 실습**
  - **<실습:gg-11-라플라시안_필터.ipynb>**

이진 영상 처리

이진 영상 처리

- 기본 이진화 : 하나의 임계값을 전체 영상에서 사용
- 적응적 이진화
  - 영상의 각 영역에 따라서 서로 다른 임계값을 사용
    - 영상 안에서도 조명 조건이 달라지는 경우에 하나의 임계값을 사용하는 것이 좋지 않을 수 있음
  - 전체 영상의 히스토그램을 이용하는 것이 아니라 영상의 일부분에 대한 히스토그램을 가지고 그 일부분만을 위한 임계값을 계산
  - 제일 간단한 방법은 아래 그림과 같이 전체 영상을 mxm개의 소영상으로 분할한 다음 각 소영상에 대한 히스토그램을 조사하여 그 소영상에 대한 임계값 $T_{ij}(1 \le i,j \le m)$ 결정



이진 영상 처리

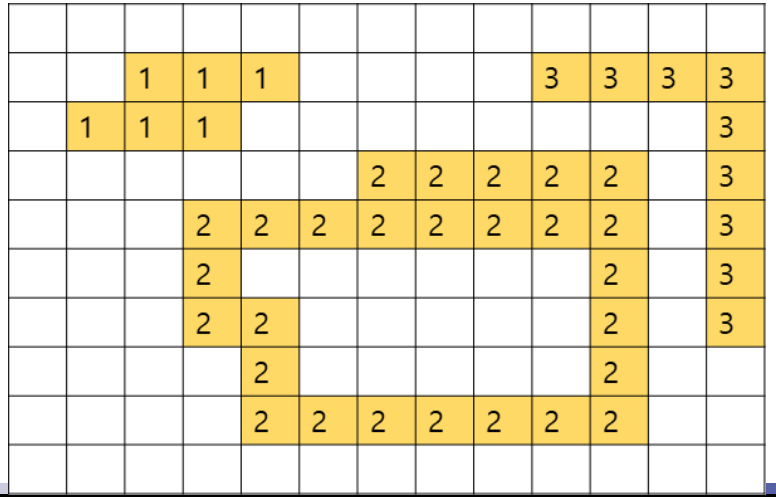
- 임계값을 찾는 방법
  - Chow와 Kaneko 접근 방식과 지역 임계값 방식 두가지
  - 2가지 방법의 가정은 영상 영역이 작을수록 균일한 조명을 가질 가능성이 높으며 임계화에 적합하다는 것이다.
- Chow와 Kaneko 방식
  - 영상을 하위 영상의 배열로 나눈 후에 각 하위 영상의 히스토그램을 조사하여 최적의 임계갑을 찾는다.
  - 각 단일 화소에 대한 임계값은 하위 영상의 결과를 보간하여 찾는다.
  - 단점은 계산 비용이 많이 들기 때문에 실시간 응용 프로그램에 부적합
- 지역 임계값 방법
  - 각 화소의 인접 화소들의 값을 통계적으로 조사하는 것
  - 입력 영상에 따라 가장 적합한 통계 모델이 달라짐
  - 많이 사용되는 수식은 다음과 같다.
  - T=인접 화소들의 평균값(mean);
  - T=인접 화소들의 중간값(median);
  - T=(max+min)/2;

실습

- **OpenCV 적응적 이진화 실습**
  - **<실습:gg-12-적응적_이진화.ipynb>**

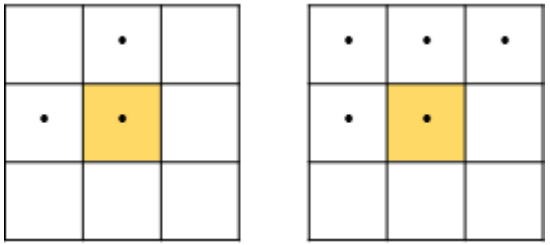
연결 성분

- 연결 성분 레이블링
  - 이진화를 통해 배경과 물체는 분리가 되었지만 물체는 모두 검은색으로 만 되어 있어서 물체가 몇 개인지 알 수 없음
  - 영상에서 서로 연결된 성분이 몇 개인지를 분석 필요
  - 영상을 스캔하여 화소 연결성을 기반으로 화소를 그룹화
  - 예를 들어 연결 성분의 모든 화소는 동일한 화소값을 공유하며 어떤 식으로든 서로 연결되어 같은 성분에 속하는 화소에 같은 레이블을 할당하고, 다른 연결 성분에는 서로 다른 레이블을 할당
  - 영상에서 연결 성분의 추출 및 레이블링은 많은 자동화된 영상 분석 프로그램의 핵심적인 부분



연결 성분

- 문제점
  - 연결 성분 레이블링 알고리즘은 영상 안의 모든 연결 성분을 찾고 같은 성분에 속하느 모든 화소에 유일한 레이블을 부여
  - 연결 성분 레이블링은 이진 영상 처리 시스템에서 상당한 시간이 걸리는 보틀넥이 되어 왔다. 왜냐하면 연결 성분을 찾는 연산은 전역 연산이고 따라서 본질적으로 순차적인 알고리즘이기 때문
- 연결 성분 레이블링 알고리즘
  - 영상 안에 하나의 물체만 있다면 연결 성분을 찾을 필요가 없을지도 모르지만, 많은 물체가 존재하고 물체의 특성과 위치가 필요하다면 연결 성분 레이블링이 수행되어야 한다.
  - 연결 성분을 찾을 때 이웃은 4-연결과 8-연결 가능

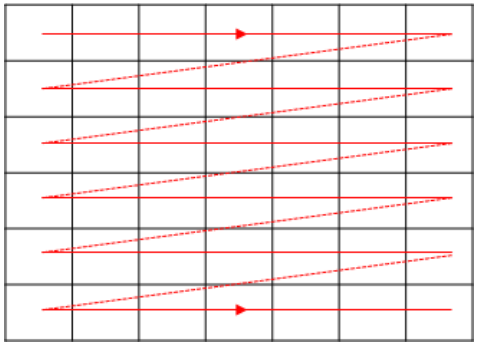


재귀 알고리즘

- 그래프 탐색 기반 알고리즘
  - 연결 성분의 첫 번째 화소를 찾으면 이 화소와 연결된 모든 화소에 레이블이 지정된다.
  - 재귀 알고리즘은 하나의 CPU만을 가진 순차 컴퓨터에서는 매우 비효율적이지만 병렬 컴퓨터에서는 많이 쓰인다.
- 재귀 연결 성분 알고리즘의 순서
  - 1. 영상을 스캔하여 레이블링되어 있지 않은 전경 화소를 찾아 새로운 레이블 L을 부여한다.
  - 2. 재귀적으로 레이블 L을 모든 이웃의 전경 화소(4-이웃 또는 8-이웃)에 부여한다.
  - 3. 더 이상 레이블링되어 있지 않은 전경 화소가 없으면 멈춘다.
  - 4. 단계 1.로 간다.

2-패스 알고리즘

- 4-연결을 이용한 2-패스 연결 성분 레이블링 알고리즘
  - 2-패스 알고리즘은 순차적 알고리즘 중 하나로, 영상을 2번 스캔
    - 첫 번째 패스에서는 연결 성분에 임시 레이블을 부여하며 화소의 이웃을 조사하여 이미 사용 중인 레이블을 부여하기 위해 노력한다.
    - 두 번째 패스에서 등가 레이블의 가장 작은 레이블로 각 임시 레이블을 대체
  - 이 알고리즘은 영상 2개 행만을 가지고 작업하므로 컴퓨터의 메모리가 작아 전체 영상을 불러올 수 없는 경우에도 사용할 수 있다.

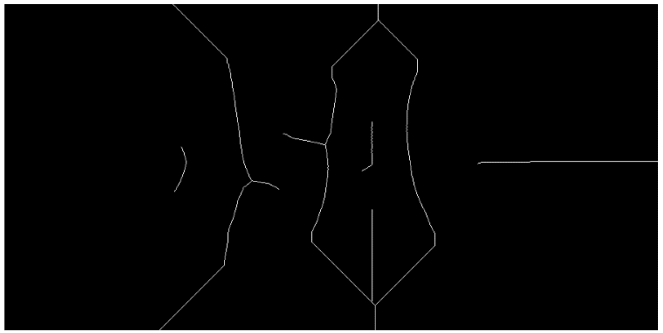


  - 첫 번째 패스
    - 1. 영상을 위에서 아래로, 왼쪽으로 오른쪽으로 스캔
    - 2. 현재 화소가 1이면
      - (a)왼쪽 화소만이 레이블을 가지면 그 레이블을 현재 화소에 부여
      - (b)왼쪽 화소만이 레이블을 가지면 그 레이블을 현재 화소에 부여
      - (c) 왼쪽과 왼쪽 화소가 다른 레이블을 가지면 이 사실을 등가 테이블에 기록
      - (d) 위의 경우가 아니면 이 화소에 새로운 레이블을 부여
    - 3. 고려해야 할 더 이상의 화소가 없으면 멈춘다.
  - 두 번째 패스
    - 1. 등가 테이블에서 각 등가 레이블 집합에서 최소의 레이블을 찾는다.
    - 2. 영상을 조사하여 레이블을 등가 집합의 최소 레이블로 바꾼다

세선화

- 이진화된 영상에서 골격을 찾는 영상처리 기법
  - 두꺼운 영역이 포함된 영상을 입력으로 받아 한 화소 두께의 출력 영상
  - 문자 인식, 지문인식, 물체인식 등의 특징추출 전 단계에 적용되는 기본적인 영상처리 알고리즘
- 골격선
  - 폭은 1이여야 하며 골격선의 위치는 선 도형의 중심에 위치해야 함
  - 골격선은 원래의 도형에 있어서 연결성을 유지해야 하며 세선화 과정에서 골격선의 길이가 계속해서 줄어들어서는 안된다.
  - 패턴 윤곽선의 작은 요철 모양이 골격선에 첨가되서는 안된다.
- 예) 아래 그림은 왼쪽의 이미지를 세선화를 하여 오른쪽 이미지의 결과를 얻은 것

DSAC



실습

- **OpenCV 연결성분 찾기 실습**
  - **<실습:gg-13-연결_성분_찾기.ipynb>**

인식, 분류, 검출의 활용

영상 검색 (Image Retrieval)

- 영상 검색 시스템
  - 클라우드에서 이미지를 검색하는 기능 및 검색 컴퓨터 시스템 데이터베이스 의 디지털 이미지 구축
- 이미지 검색 방법
  - 주석(caption) 단어를 통해 검색을 수행 할 수 있도록 캡션, 키워드, 제목 또는 설명 과 같은 메타 데이터 를 이미지에 추가
  - 수동 이미지 주석은 시간이 많이 걸리고 힘들고 비쌈
  - 자동 이미지 주석 에 대한 많은 연구가 진행
  - 소셜 웹 애플리케이션 및 시맨틱 웹 증가 및 여러 웹 기반 이미지 주석 도구의 개발에 영감
- 이미지 검색의 특징
  - 사용자는 키워드, 이미지 파일/링크와 같은 쿼리 용어를 제공
  - 시스템은 이미지와 "유사한"이미지를 쿼리에 반환
  - 검색 기준에 사용된 유사성은 메타 태그, 이미지의 색상 분포, 영역/모양 속성 등
  - 이미지 검색 시스템 설계 시 이미지 데이터의 범위와 특성 이해 중요
  - 디자인은 또한 사용자 기반의 다양성 및 검색 시스템에 대한 예상 사용자 트래픽 과 같은 요인에 의해 크게 영향을 받는다

얼굴 검출 (Face Detection) API

- 얼굴 인식 API 종류
  - 현재 Luxand cloud와, Google Cloud에서도 제공
  - 구글 모바일 머신러닝 SDK인 MLKit의 얼굴 인식 API를 사용하여 Firebase를 통해 이미지에서 얼굴을 인식하고 주요 얼굴 특징을 식별하여 인식된 얼굴에서 윤곽 추출 가능
- 얼굴 인식 API 기능
  - 셀카 및 인물 사진 꾸미기, 사용자 사진에서 아바타 생성 등의 작업 수행
  - MLKit에서 실시간으로 얼굴 인식을 수행할 수 있기 때문에 플레이어의 표정에 반응하는 게임이나 영상 채팅과 같은 애플리케이션에서 사용 가능

그림 6-24 얼굴 데이터베이스와 평균

얼굴 인식 (Face Recognition)

- 얼굴 인식 시스템
  - 사람 인식 중에서 가장 많이 연구된 기술
  - 개인 디지털 이미지 또는 비디오 프레임을 소스로 주어진 이미지에서 선택한 얼굴 특징을 데이터베이스 내의 얼굴과 비교하여 동작
  - 사람의 얼굴 질감과 모양을 기반으로 패턴을 분석하여 사람을 고유하게 식별할 수 있는 생체 인식 인공지능 기반 응용 프로그램
  - 모바일 플랫폼과 로봇 공학과 같은 다른 형태의 기술에서 더 많이 사용
  - 일반적으로 보안 시스템에서 접근 제어로 사용하며 지문, 홍채 인식과 같은 다른 생체 인식과 비교하여 정밀도는 낮지만 비접촉, 비 침습적 방법으로 널리 채택
  - 기타 응용 분야로는 고급 인간-컴퓨터 상호 작용, 비디오 감시,이미지 자동 색인 생성 및 비디오 데이터베이스
- 열굴인식 기술 (zhao2003 참고)
  - 영상 전체에서 특징을 추출하고 분류하는 통합 접근방법
    - 고유 얼굴은 얼굴 영상을 구성하는 모든 화소를 PCA로 변환한 특징 벡터를 사용
  - 영상의 특정 부분에서 특징을 추출하고 분류하는 특징 기반 접근 방법
    - 1970년대 초에 시작되어 대부분 눈, 코, 입 등에 해당하는 특징을 추출한 후 분류
    - 합성곱 신경망을 이용한 방법
  - 두 방법을 결합하는 혼합 접근 방법
  - 최근에는 깊이 영상을 이용한 얼굴 인식이 활발히 시도[Queirolo2010]