

CH01 영상처리 시작하기

Joonki Paik

Image Processing and Intelligent Systems Laboratory
Chung-Ang University
paikj@cau.ac.kr

C2-002 영상처리기초



Table of Contents

1 영상처리 시작하기

2 Homework

3 참고문헌



Table of Contents

1 영상처리 시작하기

2 Homework

3 참고문헌



학습내용

디지털 영상처리 이론 참고문헌 [1].

본 강의 주교재 : [2]

본 강의에서 사용되는 Python 코드 : [3]

- 영상처리의 정의
- 영상처리 파이프라인
- 영상처리에 사용되는 Python 라이브러리 설치 및 사용 방법
- Python을 사용한 영상 I/O 및 디스플레이
- 영상의 유형, 파일 형식 및 기본 조작 방법



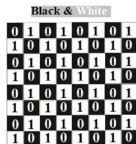
영상의 정의

2차원 영상 :

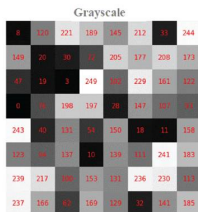
$$f(x, y, c) \quad (1)$$

(x, y) 는 가로/세로 방향의 좌표, f 는 해당 좌표의 밝기값, c 는 컬러채널 RGB 중의 하나

Single Channel

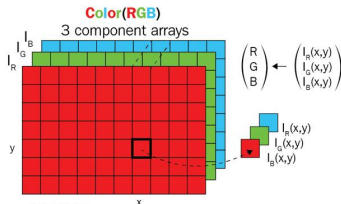


1 bit per pixel.
Values 0 ~ 1
Height x Width



8 bits per pixel.
Values 0 ~ 255
Height x Width

Multi Channel



24 bits per pixel.
RB Red/Green/Blue
Height x Width X 3:



2진 영상, 명암도 영상, RGB 영상

Binary

Hands-on Image
Processing with
Python

Gray Scale



RGB



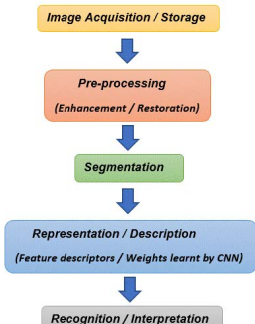
영상처리 (Image Processing)

영상처리의 정의 : 컴퓨터 알고리즘(코드)을 사용해서 영상을 조작, 분석, 해석하는 과정

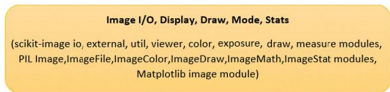
영상처리의 종류 : 복원, 표현, 분할, 분류, 감지, 해석, 인식

영상처리의 응용 : TV, 사진, 로봇시각, 원격탐사, 의료진단, 산업검사

The Image Processing Pipeline: Steps in Image Processing



Various Modules for Image Processing Tasks



Python 및 영상처리 라이브러리 설치 I

■ Anaconda Individual Edition 다운로드 설치

- <https://www.anaconda.com> => Download => 모든 옵션 디폴트로 설치 (초기 설치 시 권장)

■ Jupyter Notebook 시작폴더 지정

- 1 Windows에서 작업에 사용할 폴더 생성 : (예) C:\workspace
- 2 Anaconda Navigator (graphical user interface) 실행 > Home > base(root) > Powershell prompt 실행 :
(base) jupyter notebook --generate-config > "Writing default config to :
C:\users\name\.jupyter\jupyter_notebook_config.py."
- 3 jupyter_notebook_config.py 파일 편집 (메모장 사용) >
#c:NotebookApp.notebook_dir=' ' 찾아서 # 삭제하고 원하는 폴더 기재 >
c:NotebookApp.notebook_dir=' C:\workspace'
- 4 jupyter notebook 속성 편집 : %USERPROFILE%/ 및 %HOMEPATH% 삭제 :
Windows 시작버튼 > Jupyter Notebook 탐색 > 파일위치 열기 > Jupyter Notebook 속성편집 > 대상 : %USERPROFILE%/ 삭제, 시작위치 : %HOMEPATH% 삭제



Python 및 영상처리 라이브러리 설치 II

■ 가상환경 (p37) 생성 :

- Anaconda Navigator > Environments > Create > Name : p37 > Python : 3.7.13 > create
- Anaconda Navigator > Home > p37 > Powershell Prompt 및 Jupyter Notebook 설치

■ 가상환경에 ipykernel 추가 :

- Anaconda Navigator > Home > p37 > Powershell prompt 실행
(p37) python -m ipykernel install --user --name p37 --display-name p37

■ 가상환경에 라이브러리 설치 :

Anaconda Navigator > Home > p37 > Powershell prompt 실행

- 1 (p37) conda install -c anaconda numpy
- 2 (p37) conda install -c anaconda scipy
- 3 (p37) conda install -c anaconda scikit-image
- 4 (p37) conda install -c anaconda scikit-learn
- 5 (p37) conda install -c anaconda pillow
- 6 (p37) conda install -c simpleitk simpleitk
- 7 (p37) conda install -c conda-forge opencv
- 8 (p37) conda install -c conda-forge matplotlib

■ 가상환경에서 Jupyter Notebook 실행하기

Anaconda Navigator > Home > p37 > Jupyter Notebook 실행

- Internet browser(Chrome)에서 미리 설정한 작업폴더(C:\workspace)에 있는 ipynb 파일을 열거나, 새 파일을 생성해서 작업



Python으로 영상 읽기, 저장, 및 표시하기-실습

- PIL
- Matplotlib
- scikit-image



다양한 영상 유형과 파일 형식-실습

- 파일 형식 및 상호 변환 : BMP(8, 24, 32비트), PNG, JPG, GIF, PPM, PNM, TIFF
- 영상 유형(모드) 및 상호 변환
 - 1 단일 채널 영상 : 이진 영상 (1비트), 명암도 영상 (8비트, 0-255)
 - 2 다중 채널 영상 : 3채널 (RGB, HSV 등), 4채널 (RGBA, A는 투명도)
- 색상공간 및 상호 변환 : RGB, HSV, XYZ, YUV, YIQ, YPbPr, YCbCr, YDbDr
- 영상저장 데이터 구조 및 상호 변환
 - 1 PIL : Image 객체 사용
 - 2 scikit-image : numpy ndarray 사용



영상의 기본적인 조작-실습

■ Numpy 배열 슬라이싱으로 영상 영역 자르기



■ 교차 디졸브를 사용한 두 영상의 알파 블렌딩

$$f_3(x, y) = (1 - \alpha)f_1(x, y) + \alpha f_2(x, y), \text{ for } \alpha = 0, 0.05, 0.10, \dots, 1.00 \quad (2)$$



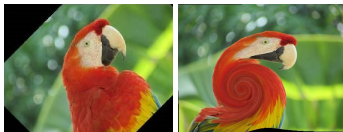
PIL을 이용한 영상 조작-실습

- 영상자르기
- 영상 크기 확대 및 축소
- 영상 반전
- 컬러 영상을 명암도 영상으로 변환
- 밝기값 변환 (intensity transformation) : 로그 변환, 파워-로(power-law) 변환
- 기하학적 변환 : 영상 반사, 영상 회전, 어파인(affien) 변환, 원근(perspective) 변환
- 영상에 캡션 추가
- 영상에 그리기 : 도형, 텍스트
- 섬네일 생성
- 기본 통계 계산 : 밝기값의 평균, 미디언, 표준 편차 등
- 히스토그램 그리기
- RGB 채널 분리
- 다중 채널 결합
- α -블렌딩 (2)
- 두 영상 더하기 (super-imposing)
- 두 영상 차분 계산



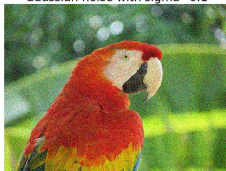
scikit-image를 사용한 영상 조작-실습

- `warp()` 함수를 사용한 기하 변환과 역 와핑 : 어파인 변환, `swirl` 변환

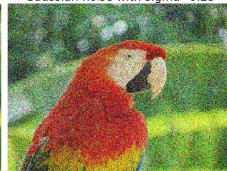


- 영상에 랜덤 가우시안 잡음 추가

Gaussian noise with sigma=0.1



Gaussian noise with sigma=0.25



Gaussian noise with sigma=0.5



Gaussian noise with sigma=1



Matplotlib을 사용한 영상 조작-실습

matplotlib 라이브러리의 pylab 모듈을 사용한 영상의 윤곽선 그리기

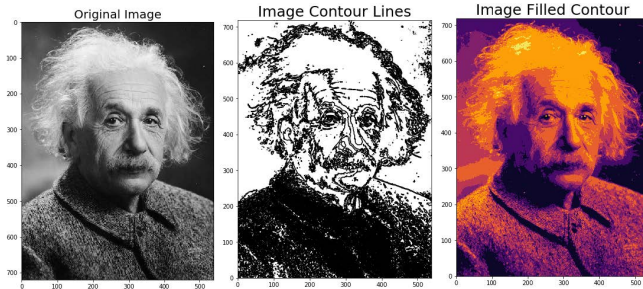


Table of Contents

1 영상처리 시작하기

2 Homework

3 참고문헌



Homework 01

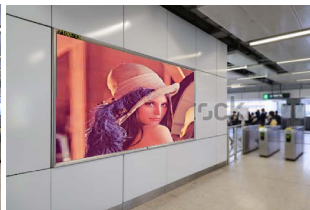
- 1 Gotham Instagram 필터 [4]의 파이썬 리메이크를 만드시오. 힌트 : 채널 보간 [5]을 만들기 위해, PIL `split()`, `merge()` 및 `numpy interp()`를 사용해서 영상을 처리
- 2 scikit-image의 `warp()` 함수를 사용하여 swirl 변환을 구현하시오. 힌트 : `swirl` 함수는 아래의 함수로 사용

```
1 def swirl(xy, x0, y0, R):
2     r = np.sqrt((xy[:,1]-x0)**2 + (xy[:,0]-y0)**2)
3     a = np.pi*r / R
4     xy[:, 1] = (xy[:, 1]-x0)*np.cos(a) + (xy[:, 0]-y0)*np.sin(a) + x0
5     xy[:, 0] = -(xy[:, 1]-x0)*np.sin(a) + (xy[:, 0]-y0)*np.cos(a) + y0
6     return xy
```

- 3 scikit-image의 transform 모듈의 `ProjectiveTransform` 함수를 사용하여, 소스에서 대상 영상으로의 사영 행렬(homography matrix)를 추정하고, `inverse()` 함수를 사용하여 다음과 같이 lena 영상을 빈 캔버스에 포함하시오.



www.shutterstock.com • 146430692



www.shutterstock.com • 146430692



Table of Contents

1 영상처리 시작하기

2 Homework

3 참고문헌



참고문헌

- [1] R. Gonzales and R. Woods, *Digital image processing 4th edition*.
Pearson, 2018.
- [2] S. Dey, *Hands-On Image Processing with Python : Expert techniques for advanced image analysis and effective interpretation of image data*.
Packt Publishing Ltd, 2018.
- [3] “Hands-on-image-processing-with-python.” <https://github.com/PacktPublishing/Hands-On-Image-Processing-with-Python>.
Accessed : 2022-07-16.
- [4] “Cv-instagram-filters.”
<https://github.com/lukeyxyz/CV-Instagram-Filters>.
Accessed : 2022-07-16.
- [5] M. Pratusевич, “Instagram filters in 15 lines of python.”
<https://www.youtube.com/watch?v=otLGdpBg1EA>, 2017.

