[RT704] Advanced Medical Image Processing

< Project >

(~ 2021-12-17)

Robotics Engineering

202123008 Jinmin Kim

Phone: 010-6266-6099

Mail: rlawlsals@dgist.ac.kr

Problem #1

Assignment 1 Image Enhancement와 관련된 논문 3개를 조사하였다.

[1] Z. Yu and C. Bajaj, "A fast and adaptive method for image contrast enhancement," Int. Conf. Image Process. ICIP, 2004

논문 요약:

- 본 논문에서는 영상 contrast enhancement를 위한 빠른 접근을 제안하였다.
- 국소 영역에 대한 contrast 조작을 통해 영상을 enhance 시키는 빠른 접근을 제안하였는데, 제안한 방법은 빠르며, 구현하기 쉽고, adaptive, multiscale, weighted localization 등의 우수한 속성을 가진다.
- 3가지 종류의 의료 영상에 대해서 본 논문에서 제안한 방법의 성능과 multiscale 속성에 대해서 설명하고 있다.
- 본 논문에서 제안한 방법은 의료 영상이 아닌 여러 종류의 영상들에 대해서도 계산 속도 및 화질 향상의 측면에서 좋은 성능의 결과를 제공할 것이다.

[2] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142-3155, 2017

논문 요약:

- 본 논문에서는 noisy image로부터 noise를 분류하기 위해 residual learning을 사용한, image denoising을 위한 CNN을 제안한다.
- 기존의 CNN과 달리 Gaussian denoising을 위한 residual image를 예측한다.
- Gaussian denoising 외의 JPEG deblokcing 등의 다른 노이즈에 대해서도 성능이 뛰어나다.
- 특정 noise level에 대해서만 학습하는 기존의 모델들과 달리, DnCNN은 한 모델로 여러 noise scale에 대한 blind Gaussian denoising을 할 수 있다.

[3] A. Sameh Arif, S. Mansor, and R. Logeswaran, "Combined bilateral and anisotropic-diffusion filters for medical image de-noising," 2011 IEEE Student Conf. Res. Dev. SCOReD 2011, pp. 420-424, 2011

논문 요약:

- 본 논문은 Image Denoising에 대해서 다룬다.
- 노이즈 제거 기술의 대표적 비선형 방법인 Bilateral filter와 Anisotropic filter를 조합해서 이미지의 선명도를 유지하면서 영상의 노이즈를 줄일 수 있는 방법을 제안하였다.

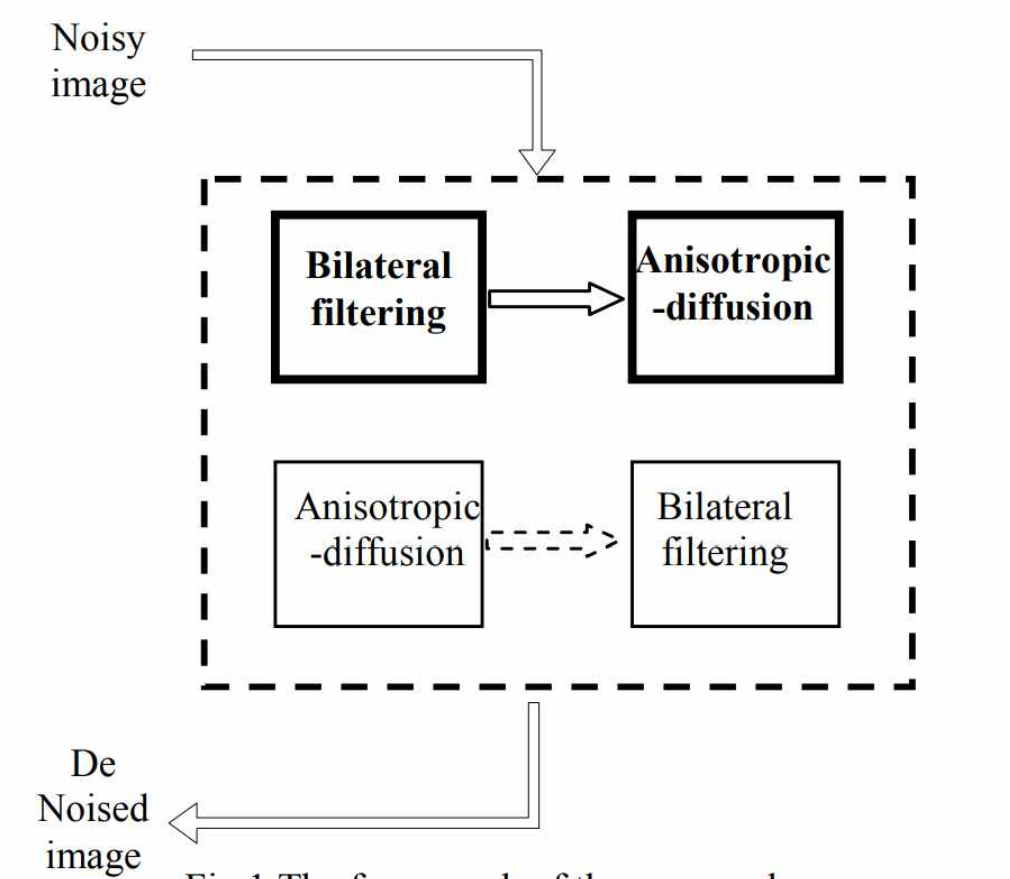


Fig.1.The framework of the proposed (in bold) and alternativemethods

- 결과적으로 Bilateral -> Anisotropic filter 순서로 조합했을 때 가장 좋은 결과를 얻었다.
- 23개의 MRI 이미지로 실험하였고, 제안된 방법을 통해 얻은 결과는 기존 방법에 비해 8% 이상 높은 PSNR(peak signal to noise ratio) 및 CR(compression ratio) 성능을 얻었다.

Problem #2

Problem #1에서 조사한 논문 중 [3] “Combined bilateral and anisotropic-diffusion filters for medical image de-noising”을 구현하여 Assignment 1에서 구현한 것과 PSNR score를 비교하였다.

- 논문의 Bilateral filter와 Anisotropic diffusion filter를 구현하기 위해 MATLAB의 함수를 이용하여 프로그래밍하였다.
- 기존 Assignment 1에서 Python으로 구현한 코드는 MATLAB으로 재구현하여 두 결과를 비교할 수 있도록 하였다.

- 대상 이미지의 경우 IMA파일이라 MATLAB에서 제공하는 dicomread 함수로 직접 읽어올 수 없기때문에 Python에서 불러온 IMA파일의 array를 jpg 파일로 저장하여 사용하였다.

- **파일 => “Project.m” 참조**
- **소스 이미지 파일 : “NDCT image.jpg”, “LDCT image.jpg”**

1) 코드 설명

```
1  clear all;
2  close all;
3  clc;
4
5  %% 데이터 로드 및 변환전 PSNR 계산
6  image_NDCT=rgb2gray(double(imread('NDCT image.jpg'))/255);
7  image_LDCT=rgb2gray(double(imread('LDCT image.jpg'))/255);
8
9  % PSNR score 출력 ( 변환전LDCT 영상 vs NDCT 영상 )
10 psnr_LDCT_NDCT = psnr(image_LDCT, image_NDCT)
11
12
13 % NDCT image 플롯
14 figure(1);
15 subplot(2,5,1);
16 imshow(image_NDCT);
17 title({'NDCT image';'(Ground-truth)'},'fontsize',16);
18
19
20 % Original LDCT 플롯
21 subplot(2,5,2);
22 imshow(image_LDCT);
23 title({'Original LDCT';'(Noised image)';
24     ['#color{red}PSNR score : ', num2str(psnr_LDCT_NDCT)]},'fontsize',16);
25
26
27
```

1~27 : 데이터 로드 및 변환전 LDCT, NDCT 이미지 간의 PSNR 계산

```
28    %% 논문 구현
29
30
31    %% Anisotropic filter만 사용 (1)
32 -  image_Aniso = image_LDCT;
33
34    % Anisotropic Diffusion Filtering 구현
35 -  image_Aniso = imdiffusefilt(image_Aniso);
36
37    % PSNR score 출력 ( Anisotropic-diffusion filtering 후 LDCT 영상 vs NDCT 영상 )
38 -  psnr_A_NDCT = psnr(image_Aniso, image_NDCT);
39
40    % Anisotropic-diffusion filtering 후 LDCT 영상 플롯
41 -  subplot(3,4,5);
42 -  imshow(image_Aniso);
43 -  title({'(1) Anisotropic-Diffusion Filtering';
44        ['₩color{red}PSNR score : ', num2str(psnr_A_NDCT)]},'fontweight','bold', 'fontsize',16);
45
46
47    %% Bilateral filter만 사용 (2)
48 -  image_Bilat = image_LDCT;
49
50    % Bilateral Filtering 구현
51 -  image_Bilat = imbilatfilt(image_Bilat);
52
53    % PSNR score 출력 ( Bilateral filtering 후 LDCT 영상 vs NDCT 영상 )
54 -  psnr_B_NDCT = psnr(image_Bilat, image_NDCT);
55
56    % Bilateral filtering후 LDCT 영상 플롯
57 -  subplot(3,4,6);
58 -  imshow(image_Bilat);
59 -  title({'(2) Bilateral Filtering';
60        ['₩color{red}PSNR score : ', num2str(psnr_B_NDCT)]},'fontweight','bold', 'fontsize',16);
61
62
63    %% Bilateral -> Anisotropic 순서로 사용 (3)
64 -  image_Bilat_Aniso = image_LDCT;
65
66    % Bilateral Filtering 구현
67 -  image_Bilat_Aniso = imbilatfilt(image_Bilat_Aniso);
68
69    % Anisotropic Diffusion Filtering 구현
70 -  image_Bilat_Aniso = imdiffusefilt(image_Bilat_Aniso);
71
72    % PSNR score 출력 ( Bilateral -> Anisotropic 후 LDCT 영상 vs NDCT 영상 )
73 -  psnr_BA_NDCT = psnr(image_Bilat_Aniso, image_NDCT);
74
75    % Bilateral -> Anisotropic 후 LDCT 영상 플롯
76 -  subplot(3,4,7);
77 -  imshow(image_Bilat_Aniso);
78 -  title({'(3) Bilateral -> Anisotropic-Diffusion Filtering';
79        ['₩color{red}PSNR score : ', num2str(psnr_BA_NDCT)]},'fontweight','bold', 'fontsize',16);
80
81
82    %% Anisotropic -> Bilateral 순서로 사용 (4)
83 -  image_Aniso_Bilat = image_LDCT;
84
85    % Anisotropic Diffusion Filtering 구현
86 -  image_Aniso_Bilat = imdiffusefilt(image_Aniso_Bilat);
87
88    % Bilateral Filtering 구현
89 -  image_Aniso_Bilat = imbilatfilt(image_Aniso_Bilat);
90
91    % PSNR score 출력 ( Anisotropic -> Bilateral 후 LDCT 영상 vs NDCT 영상 )
92 -  psnr_AB_NDCT = psnr(image_Aniso_Bilat, image_NDCT);
93
94    % Anisotropic -> Bilateral 후 LDCT 영상 플롯
95 -  subplot(3,4,8);
96 -  imshow(image_Aniso_Bilat);
97 -  title({'(4) Anisotropic-Diffusion -> Bilateral Filtering';
98        ['₩color{red}PSNR score : ', num2str(psnr_AB_NDCT)]},'fontweight','bold', 'fontsize',16);
99
100
```

28~100 : Bilateral Filter와 Anisotropic-diffusion Filter를 구현

=> MATLAB의 imilatfilt 함수와 imdiffusefilt 함수를 통해 필터를 각각 구성함.

(1) : Anisotropic-diffusion Filter만 사용

(2) : Bilateral Filter만 사용

(3) : Bilateral Filter -> Anisotropic-diffusion Filter 순서로 사용

(4) : Anisotropic-diffusion Filter -> Bilateral Filter 순서로 사용

```
%% Assignment 1


%% Box filtering (a)

% Box filtering
filterSize = 3;
image_a = imboxfilt(image_LDCT, filterSize);

% PSNR score 출력 ( Box filtering 후 LDCT 영상 vs NDCT 영상 )
psnr_Box_NDCT = psnr(image_a, image_NDCT);

% Box filtering 후 영상 플롯
subplot(3,4,9);
imshow(image_a);
title({'(a) Box Filtered image of LDCT';
    ['\color{red}PSNR score : ', num2str(psnr_Box_NDCT)]},'fontsize',16);


%% Gaussian filtering (b)

% Gaussian filtering
sigma = 0.5;
image_b = imgaussfilt(image_LDCT, sigma);

% PSNR score 출력 ( Gaussian filtering 후 LDCT 영상 vs NDCT 영상 )
psnr_Gaussian_NDCT = psnr(image_b, image_NDCT);

% Gaussian filtering 후 영상 플롯
subplot(3,4,10);
imshow(image_b);
title({'(b) Gaussian Filtered image of LDCT';
    ['\color{red}PSNR score : ', num2str(psnr_Gaussian_NDCT)]},'fontsize',16);



%% Sharpening filtering (c)

% Sharpening filtering
image_c = imsharpen(image_LDCT,'Radius',2,'Amount',1);

% PSNR score 출력 ( Sharpening filtering 후 LDCT 영상 vs NDCT 영상 )
psnr_Sharpening_NDCT = psnr(image_c, image_NDCT);

% Sharpening filtering 후 영상 플롯
subplot(3,4,11);
imshow(image_c);
title({'(c) Sharpening Filtered image of LDCT';
    ['\color{red}PSNR score : ', num2str(psnr_Sharpening_NDCT)]},'fontsize',16);



%% Median filtering (d)

% Median filtering
kernel = [7 7];
image_d = medfilt2(image_LDCT,kernel);

% PSNR score 출력 ( Median filtering 후 LDCT 영상 vs NDCT 영상 )
psnr_Median_NDCT = psnr(image_d, image_NDCT);

% Median filtering 후 영상 플롯
subplot(3,4,12);
imshow(image_d);
title({'(d) Median Filtered image of LDCT';
    ['\color{red}PSNR score : ', num2str(psnr_Median_NDCT)]},'fontsize',16);
```

101~168 : Assignment 1의 코드를 MATLAB에서 재구현

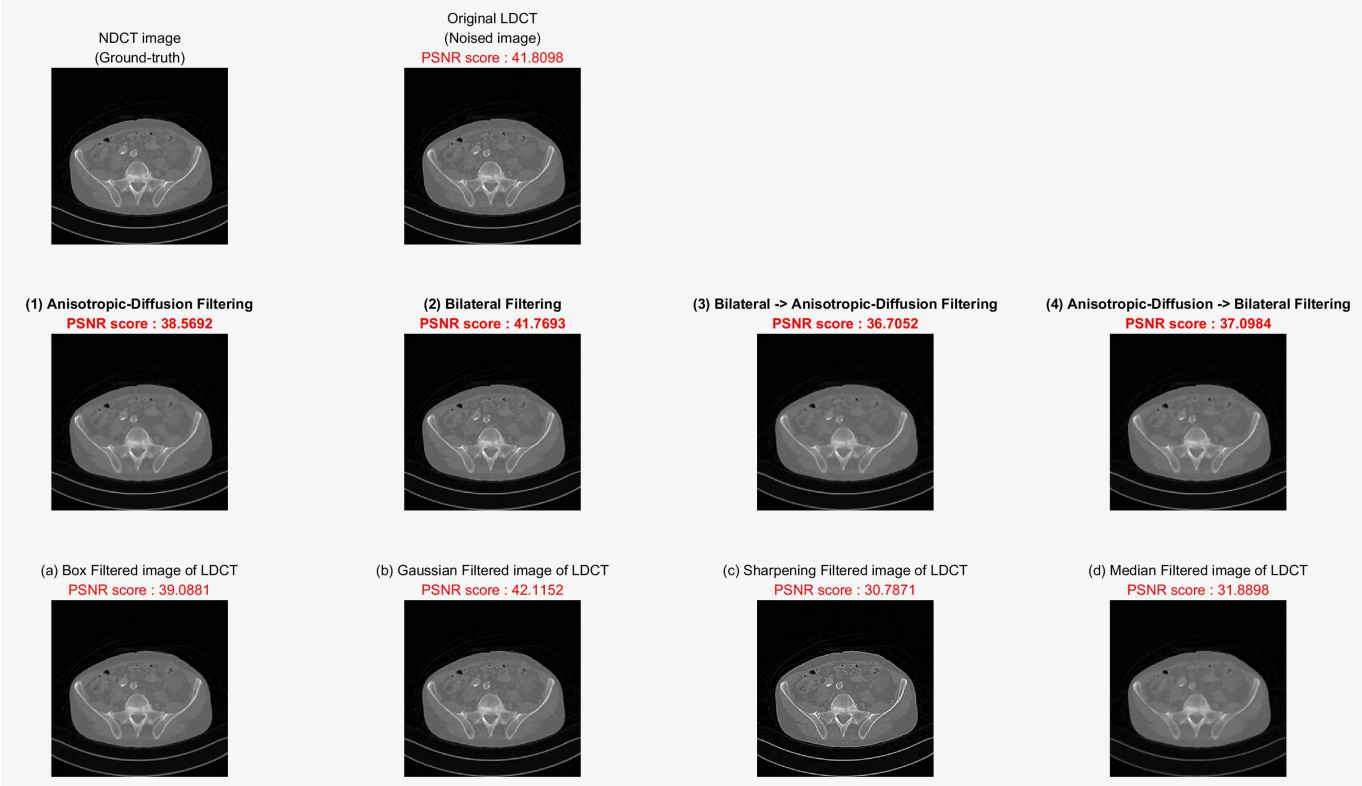
(a) : Box filtering

(b) : Gaussian filtering

(c) : Sharpening filtering

(d) : Median filtering

2) 결과



1행 : NDCT image와 필터링 전 LDCT 이미지
2행 : 논문에서 제안한 방법
3행 : Assignment 1에서 진행했던 방법

=> PSNR score는 각 그림 부제목에 적색으로 기입하였다.

3) 고찰

논문에서 제시한 방법에 의하면, (3)의 결과가 가장 좋게 나올 것으로 예상하였다. 하지만 2행 중에서 가장 결과가 안좋은 것이 (3)이었고 가장 결과가 좋은 것은 Bilateral filter만을 사용한 (2)의 결과였다.

원인을 파악해보자면 우선 MATLAB으로 간단하게 구현된 코드이므로 Filter의 파라미터를 모두 정확하게 반영하지 못했을 가능성이 있다. 그리고 현재 비교하고 있는 이미지의 경우 이미지 노이즈가 그렇게 많지 않아 결과에 차이가 보이지 않을 수 있다.

Bilateral filter는 비반복적 로컬 접근법으로서, 필터링된 영상은 각 필셀의 강도 값을 인접 픽셀 간의 기하학적 및 광도 유사성에 의해 가중치가 부여된 평균 값으로 대체하여 얻어진다. 수식은 다음과 같다.

$$h(x) = k^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\zeta) c(\zeta, x) s(f(\zeta), f(x)) d\zeta$$

where

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\zeta, x) f(f(\zeta), f(x)) d\zeta$$

Anisotropic-diffusion filter는 노이즈를 제거함과 동시에 엣지를 보존해주는데, 노이즈를 제거하기 위해 다음과 같은 편미분 방정식을 사용한다.

$$I_t = div(c(x, y, t)\nabla I) = c(x, y, t)\Delta I + \nabla c.\nabla I$$

두 필터 모두 대표적인 Non-linear 필터로, 저자들은 성능 향상을 위해 다른 필터들과 조합하여 사용하기를 권장하고 있다. 유사 연구들에서는 wavelet thresholding, shock filter 등과 결합하여 성능 향상을 이룬 사례가 있다.

Assignment 1에서 구현한 기본 필터들을 비롯해서 최근 연구되는 필터들과 결합한다면 이미지 노이즈 제거에서 뿐만 아니라 다른 성능 향상도 기대해 볼 수 있을 것이다.