Мета роботи: ознайомлення з методами міокардіосцинтиграфічних досліджень та принципами побудови сцинтиграфічних зображень серця. Отримання навичок дослідження планарних сцинтиграфічних зображень міокарду у середовищіMATLAB.

Програмний код:

ЧАСТИНА І

1. Надано20 сцинтиграфічних зображень, які отримані в певній фазі серцевого циклу(проміжок часу між фазами1 с). Послідовно завантажити в середовище Matlab планарні сцинтиграфічні зображення«1-20.jpg».

t = 1:20;

tnew = 1:0.1:20;

for i = 1:20

Im(:,:,i) = im2double(rgb2gray(imread(strcat('D:\Магістри\_2й семестр\БЕС\20\',num2str(i),'.jpg'))));

figure(1);subplot(2,10,i); imshow(Im(:,:,i)); title(strcat('№', num2str(i)));

end



2. На зображеннях виділити прямокутну область інтересу і розрахувати сумарну інтенсивність пікселів отриманої області. Виділена область повинна бути однакова для всіх зображень.

Im\_size = size(Im);

Im\_number = input('Введіть номер зображення (1..20):')

figure(2); imshow(Im(:,:,Im\_number));

h = imrect;

disp('Оберіть зону інтересу');

pos = getPosition(h);

for i = 1:Im\_size(3)

Im\_crop = imcrop(Im(:,:,i),pos);

intensity(i)=sum(sum(Im\_crop));

end

3. Виконати нормування і побудувати графік залежності активності міокарда від часу протягом кардіоциклу від часу.

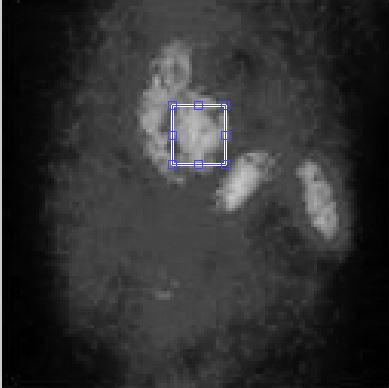
4. Порівняти отриману криву активності серцевого циклу зі стандартом і зробити висновки.

intensity\_norm = intensity./max(intensity);

intensity\_norm\_new = interp1(t, intensity\_norm, tnew, 'spline');

figure(3); plot(t,intensity\_norm,'\*',tnew,intensity\_norm\_new); title('Залежність активності міокарда протягом кардіоциклу від часу'); grid on

xlabel('t,c'); ylabel('N,к-сть імпульсів');



5. Надано в якості початкового зображення сцинтиграма серця в медичному форматіDICOM. Виконати зашумлення початкового зображення гаусівським шумом.

info = dicominfo('D:\Магістри\_2й семестр\БЕС\1.dcm');

Im\_dicom = dicomread(info);

Im\_dicom = im2double(rgb2gray(Im\_dicom));

Im\_dicom\_noise = imnoise(Im\_dicom,'gaussian');

6. Виконати просторову фільтрацію зашумленого зображення методом лінійного усереднення пікселів по їх сусіднім значенням.

mask = fspecial('average');

Im\_dicom\_filter = imfilter(Im\_dicom\_noise, mask);

figure(4); subplot(1,3,1); imshow(Im\_dicom),title('Зображення DICOM')

subplot(1,3,2),imshow(Im\_dicom\_noise),title('З гаусівським шумом')

subplot(1,3,3),imshow(Im\_dicom\_filter),title('Після фільтрації')



7. Виконати частотну фільтрацію зашумленого зображення, використовуючи пряме і обернене перетворення Фур'є з застосуванням ФНЧ і ФВЧ(фільтрів Гауса і Баттерворта).

NameOfFilters = {'gaussian','btw'};

[M N] = size(Im\_dicom\_noise);

F = fft2(Im\_dicom\_noise);

sig = 10; figure(5);

for i = 1:length(NameOfFilters)

LOW = lpfilter(strcat(NameOfFilters{i}), M, N, sig);

G\_low = LOW.\*F;

g\_low = real(ifft2(G\_low));

subplot(2,2,i),imshow(g\_low),title(strcat('ФНЧ = ',num2str(NameOfFilters{i})));

Cor\_filter\_LOW = corr2(Im\_dicom, g\_low)

HIGH = hpfilter(NameOfFilters{i}, M, N, sig);

G\_high = HIGH.\*F;

g\_high = real(ifft2(G\_high));

subplot(2,2,(i+2)),imshow(g\_high),title(strcat('ФВЧ = ',num2str(NameOfFilters{i})));

Cor\_filter\_HIGH = corr2(Im\_dicom, g\_high)

end

Cor\_average = corr2(Im\_dicom, Im\_dicom\_filter)



8. Оцінити результати фільтрації шляхом розрахування коефіцієнтів кореляції між початковим зображенням і зображеннями, отриманими після застосування різних фільтрів, та зробити висновки.

Cor\_filter\_LOW = 0.9940

Cor\_filter\_HIGH = 0.0624

Cor\_filter\_LOW = 0.9938

Cor\_filter\_HIGH = 0.0955

Cor\_average = 0.9918

ЧАСТИНА ІІ

1. Для наданого зображення сцинтиграми серця сформувати зображення, спотворені шумами таких типів:

- гаусівський білий шум зі змінною дисперсією, яка є випадковою величиною зі значеннями від0 до1;

- шум типу«сіль та перець» для20 відсотків пікселів;

2. Для наданого зображення сцинтиграми серця сформувати зображення, зашумлені періодичним шумом у вигляді двовимірної синусоїди(частоти обрати довільно):



DICOM\_gaus = imnoise(Im\_dicom, 'gaussian', rand(1), 0.03);

DICOM\_salt = imnoise(Im\_dicom, 'salt & pepper', 0.2);

A = 0.075; phi\_x = 0; phi\_y = 0; f1 = 0.04; f2 = 0.02; y = 1:N;

for x = 1:M

n(x,y)=A\*sin(2\*pi\*f1\*(x+phi\_x)./1+2.\*pi\*f2.\*(y+phi\_y)./1);

end

DICOM\_sin = imadd(Im\_dicom, n);

figure(6),subplot(1,3,1), imshow(DICOM\_gaus), title('Гаусівський білий шум')

subplot(1,3,2), imshow(DICOM\_salt), title('Шум salt & pepper');

subplot(1,3,3), imshow(DICOM\_sin), title('SIN шум')



3. Вивчити теоретичні основи вейвлет-аналізу зображень. Обрати дві довільні різні материнські вейвлет-функції зі списку:

- вейвлети Добеші;

- вейвлети Койфмана;

- симетричні вейвлети;

- дискретні вейвлети Мейера;

- біортогональні вейвлети;

- обернені біортогональні.

Побудувати графіки вейвлетів та масштабуючих функцій.

NameOfWavelets = {'sym4','db4'};

for i = 1:length(NameOfWavelets)

[phi, psi, xval] = wavefun(NameOfWavelets{i});

figure(6+i); subplot(1,2,1); plot(xval, phi); title(strcat('Wavelet function ', NameOfWavelets{i})); grid on;

subplot(1,2,2); plot(xval,psi); title(strcat('Scale function ', NameOfWavelets{i})); grid on;

end



4. Розрахувати коефіцієнти фільтрів для кожного вейвлета для рівня декомпозиції N=3. Розрахувати вектори вейвлет-коефіцієнтів для початкового зображення та зображень, отриманих в п.1-2. Отримати вектор апроксимуючих та горизонтальних, вертикальних та діагональних деталізуючих коефіцієнтів для одного зображення і одної вейвлет-функції для

рівня розкладу N=2.

images = {Im\_dicom, DICOM\_gaus, DICOM\_salt, DICOM\_sin};

for j = 1:length(images)

figure(8+j)

for i = 1:length(NameOfWavelets)

[phi, psi, xval] = wavefun(NameOfWavelets{i});

[Lo\_D, Hi\_D, Lo\_R, Hi\_R] = wfilters(NameOfWavelets{i});

[C, S] = wavedec2(images{j}, 3, Lo\_D, Hi\_D);

coef\_approx = appcoef2(C, S, Lo\_D, Hi\_D);

subplot(3,2,i), imshow(coef\_approx), title(strcat('Фільтер = ',num2str(NameOfWavelets{i})));

end

end

[Lo\_D, Hi\_D, Lo\_R, Hi\_R] = wfilters(NameOfWavelets{1});

[C, S] = wavedec2(DICOM\_salt, 2, Lo\_D, Hi\_D);

[H, V, D] = detcoef2('all', C, S, 2);

sa2 = appcoef2(C,S,Lo\_R,Hi\_R,2);

figure(13), subplot(2,2,1), imshow(sa2); title('Апроксимуючі коефіцієнти')

subplot(2,2,2), imshow(H), title('Горизонтальні коефіцієнти')

subplot(2,2,3), imshow(V), title('Вертикальні коефіцієнти')

subplot(2,2,4), imshow(D), title('Діагональні коефіцієнти')







5. Вивчити теоретичні основи порогової обробки коефіцієнтів з використанням м’якого та жорсткого порогів. Виконати порогову обробку вейвлет-коефіцієнтів для зашумлених зображень для:

- апроксимаційної частини;

- горизонтальних деталей;

- вертикальних деталей;

- діагональних деталей;

- всіх деталей;

- всіх коефіцієнтів.

6. Відновити відфільтровані зображення. Зробити висновки щодо вибору оптимального способу фільтрації для кожного виду порогу та для кожного виду шумів.

type = {'h','v','d','t'};

for j = 2:length(images)

for k = 1:length(type)

figure()

th = thselect(images{j},'minimaxi');

[C, S] = wavedec2(images{j}, 3, Lo\_D, Hi\_D);

NC\_soft = wthcoef2(type{k}, C, S, [1 2 3], [th th th], 's');

NC\_hard = wthcoef2(type{k}, C, S, [1 2 3], [th th th], 'h');

im\_rec\_soft = waverec2(NC\_soft, S, NameOfWavelets{1});

im\_rec\_hard = waverec2(NC\_hard, S, NameOfWavelets{1});

param\_noise\_soft =(images{j} - im\_rec\_soft);

param\_noise\_hard =(images{j} - im\_rec\_hard);

trash\_soft = 20\*log10(std2(images{j})/ std2(param\_noise\_soft))

trash\_hard = 20\*log10(std2 (images{j})/ std2(param\_noise\_hard))

subplot(1,2,1), imshow(im\_rec\_soft), title(strcat('im\_rec\_soft wthcoef = ',num2str(type{k})))

subplot(1,2,2), imshow(im\_rec\_hard), title(strcat('im\_rec\_hard wthcoef = ',num2str(type{k})))

end

end



7. Побудувати зображення коефіцієнтів перетворення для початкового та зашумленого зображення на екрані з використанням функції wave2gray.

for j = 1:length(images)

[C, S] = wavedec2(images{j}, 3, Lo\_D, Hi\_D);

ww = wave2gray(C, S);

figure(), imshow(ww)

end



8. Визначити відношення сигнал/шум до та після фільтрації для кожного зображення. Зробити висновки.

trash\_soft =7.3290

trash\_hard = 7.3290

trash\_soft =7.2325

trash\_hard = 7.2325

trash\_soft =7.6428

trash\_hard = 7.6428

trash\_soft =2.6300

trash\_hard =2.6300

trash\_soft = 7.5443

trash\_hard = 7.5443

trash\_soft = 7.5681

trash\_hard =7.5681

trash\_soft =7.5239

trash\_hard =7.5239

trash\_soft = 2.7663

trash\_hard = 2.7663

trash\_soft = 21.6176

trash\_hard =21.6176

trash\_soft =22.6304

trash\_hard =22.6304

trash\_soft = 26.2089

trash\_hard =26.2089

trash\_soft =18.3112

trash\_hard =18.3112