Repository file 1

Matlab programme for Method A applied on Fig. ‘a’:

```
%%load image
I=imread('1_u.jpg');
figure,imshow(I)
title('Original image')
%%Image adjust
Istrech = imadjust(I,stretchlim(I));
figure,imshow(Istrech)
title('Contrast stretched image')
%%Convert RGB image to gray
Igray_h = rgb2gray(Istrech);
figure,imshow(Igray_h,[])
title('RGB to gray (contrast stretched) ')
%%Image segmentation by thresholding
%use incremental value to run this selection till required threshold 'level' is
%achieved
level = 0.08;
gradmag = im2bw(Igray_h,level);
figure,imshow(gradmag)
title('Segmented cracks')
%%Image morphological operation
BW = bwmorph(gradmag,'clean',10);
figure,imshow(BW)
title('Cleaned image')
BW1 = bwmorph(gradmag,'thin', inf);
figure,imshow(BW1)
title('Thinned image')

% %%Image tool
% figure,imtool(BW1)
% figure,imtool(I)
%%Calaculate crack length
calibration_length=0.001;
calibration_pixels=1000;
crack_pixel=35;
crack_length=(crack_pixel *calibration_length)/calibration_pixels
```

```
crack_length =


   3.5000e-05
```

Matlab programme for Method B applied on Fig. 'a':

```
clear all;
close all;
Irgb = imread('1_u.jpg');
figure
image(Irgb,'CDataMapping','scaled')
title('Original Image')
Igray = rgb2gray(Irgb);

figure
image(Igray,'CDataMapping','scaled')
colormap('gray')
title('Input Image in Grayscale')
I = im2double(Igray);
Gx = [-1 1];
Gy = Gx';
Ix = conv2(I,Gx,'same');
Iy = conv2(I,Gy,'same');
figure
image(Ix,'CDataMapping','scaled')
colormap('gray')
title('Ix')
figure
image(Iy,'CDataMapping','scaled')
colormap('gray')
title('Iy')
edgeFIS = mamfis('Name','edgeDetection');
edgeFIS = addInput(edgeFIS,[-1 1],'Name','Ix');
edgeFIS = addInput(edgeFIS,[-1 1],'Name','Iy');
sx = 0.1;
sy = 0.1;
edgeFIS = addMF(edgeFIS,'Ix','gaussmf',[sx 0],'Name','zero');
edgeFIS = addMF(edgeFIS,'Iy','gaussmf',[sy 0],'Name','zero');
edgeFIS = addOutput(edgeFIS,[0 1],'Name','Iout');
wa = 0.1;
wb = 0.5;
wc = 1;
ba = 0;
bb = 0.5;
bc = 1;
edgeFIS = addMF(edgeFIS,'Iout','trimf',[wa wb wc],'Name','white');
edgeFIS = addMF(edgeFIS,'Iout','trimf',[ba bb bc],'Name','black');
figure
subplot(2,2,1)
plotmf(edgeFIS,'input',1)
title('Ix')
subplot(2,2,2)
plotmf(edgeFIS,'input',2)
title('Iy')
subplot(2,2,[3 4])
plotmf(edgeFIS,'output',1)
title('Iout')
r1 = "If Ix is zero and Iy is zero then Iout is white";
r2 = "If Ix is not zero or Iy is not zero then Iout is black";
edgeFIS = addRule(edgeFIS,[r1 r2]);
edgeFIS.Rules
Ieval = zeros(size(I));
for ii = 1:size(I,1)
    Ieval(ii,:) = evalfis(edgeFIS,[(Ix(ii,:));(Iy(ii,:))]');
end

figure
image(Ieval,'CDataMapping','scaled')
colormap('gray')
```

```
title('Edge Detection Using Fuzzy Logic')
```

```
Warning: Function power has the same name as a MATLAB builtin. We suggest you
rename the function to avoid a potential name conflict.


ans =


  1×2 fisrule array with properties:

    Description
    Antecedent
    Consequent
    Weight
    Connection

  Details:

                        Description
        ___________________________________________

    1    "Ix==zero & Iy==zero => Iout=white (1)"
    2    "Ix~=zero | Iy~=zero => Iout=black (1)"
```

Matlab programme for Method C applied on Fig. 'a':

```
close all;
clear all;

I1 = imread('1_u.jpg');
figure, imshow(I1)
title('Original Image');
% text(size(I,2),size(I,1)+15, ...
%     'Image courtesy of Alan Partin', ...
%     'FontSize',7,'HorizontalAlignment','right');
% text(size(I,2),size(I,1)+25, ....
%     'Johns Hopkins University', ...
%     'FontSize',7,'HorizontalAlignment','right');
I=rgb2gray(I1);
[~,threshold] = edge(I,'sobel');
figure, imshow(I)
title('Gray Image')
fudgeFactor = 0.5;
BWs = edge(I,'sobel',threshold * fudgeFactor);

figure, imshow(BWs)
title('Binary Gradient Mask')

se90 = strel('line',3,90);
se0 = strel('line',3,0);

BWsdil = imdilate(BWs,[se90 se0]);
figure, imshow(BWsdil)
title('Dilated Gradient Mask')

sigma_i=0.01;
sigma_s=1;
[filtered, masked] = myBilateralFiltering(BWsdil, sigma_i, sigma_s);
figure, imshow(filtered)
title('Bilateral Filtered Image')
```

Matlab programme for Method D applied on “Fig. a”

```
close all;
clear all;
I1 = imread('1_u.jpg');

figure, imshow(I1)
title('Original Image')

I=rgb2gray(I1);
figure, imshow(I)
title('Gray Image')


BW1 = edge(I,'sobel', 0.15);
figure, imshow(BW1)
title('Sobel')

BW2 = edge(I,'canny',[0.1 0.4]);
figure, imshow(BW2)
title('Canny')

BW3 = edge(I,'prewitt', 0.15);
figure, imshow(BW3)
title('Prewitt')

BW4 = edge(I,'roberts', 0.15);
figure, imshow(BW4)
title('Roberts')

BW5 = edge(I,'log', 0.01);
figure, imshow(BW5)
title('LoG')

BW6 = edge(I,'zerocross',0.01);
figure, imshow(BW6)
title('Zerocross')
```

Matlab programme for Method A applied on Fig. ‘b’:

```
%%load image
I=imread('25_u.jpg');
figure,imshow(I)
title('Original image')
%%Image adjust
Istrech = imadjust(I,stretchlim(I));
figure,imshow(Istrech)
title('Contrast stretched image')
%%Convert RGB image to gray
Igray_h = rgb2gray(Istrech);
figure,imshow(Igray_h,[])
title('RGB to gray (contrast stretched) ')
%%Image segmentation by thresholding
%use incremental value to run this selection till required threshold 'level' is
%achieved
level = 0.08;
gradmag = im2bw(Igray_h,level);
figure,imshow(gradmag)
title('Segmented cracks')
%%Image morphological operation
BW = bwmorph(gradmag,'clean',10);
figure,imshow(BW)
title('Cleaned image')
BW1 = bwmorph(gradmag,'thin', inf);
figure,imshow(BW1)
title('Thinned image')

% %%Image tool
% figure,imtool(BW1)
% figure,imtool(I)
%%Calaculate crack length
calibration_length=0.001;
calibration_pixels=1000;
crack_pixel=35;
crack_length=(crack_pixel *calibration_length)/calibration_pixels
```

```
crack_length =


   3.5000e-05
```

Matlab programme for Method B applied on Fig. 'b':

```
clear all;
close all;
Irgb = imread('25_u.jpg');
figure
image(Irgb,'CDataMapping','scaled')
title('Original Image')
Igray = rgb2gray(Irgb);

figure
image(Igray,'CDataMapping','scaled')
colormap('gray')
title('Input Image in Grayscale')
I = im2double(Igray);
Gx = [-1 1];
Gy = Gx';
Ix = conv2(I,Gx,'same');
Iy = conv2(I,Gy,'same');
figure
image(Ix,'CDataMapping','scaled')
colormap('gray')
title('Ix')
figure
image(Iy,'CDataMapping','scaled')
colormap('gray')
title('Iy')
edgeFIS = mamfis('Name','edgeDetection');
edgeFIS = addInput(edgeFIS,[-1 1],'Name','Ix');
edgeFIS = addInput(edgeFIS,[-1 1],'Name','Iy');
sx = 0.1;
sy = 0.1;
edgeFIS = addMF(edgeFIS,'Ix','gaussmf',[sx 0],'Name','zero');
edgeFIS = addMF(edgeFIS,'Iy','gaussmf',[sy 0],'Name','zero');
edgeFIS = addOutput(edgeFIS,[0 1],'Name','Iout');
wa = 0.1;
wb = 0.5;
wc = 1;
ba = 0;
bb = 0.5;
bc = 1;
edgeFIS = addMF(edgeFIS,'Iout','trimf',[wa wb wc],'Name','white');
edgeFIS = addMF(edgeFIS,'Iout','trimf',[ba bb bc],'Name','black');
figure
subplot(2,2,1)
plotmf(edgeFIS,'input',1)
title('Ix')
subplot(2,2,2)
plotmf(edgeFIS,'input',2)
title('Iy')
subplot(2,2,[3 4])
plotmf(edgeFIS,'output',1)
title('Iout')
r1 = "If Ix is zero and Iy is zero then Iout is white";
r2 = "If Ix is not zero or Iy is not zero then Iout is black";
edgeFIS = addRule(edgeFIS,[r1 r2]);
edgeFIS.Rules
Ieval = zeros(size(I));
for ii = 1:size(I,1)
    Ieval(ii,:) = evalfis(edgeFIS,[(Ix(ii,:));(Iy(ii,:))]');
end

figure
image(Ieval,'CDataMapping','scaled')
colormap('gray')
title('Edge Detection Using Fuzzy Logic')
```

```
ans =

  1×2 fisrule array with properties:

    Description
    Antecedent
    Consequent
    Weight
    Connection

  Details:

                        Description
          ___________________________________________

    1     "Ix==zero & Iy==zero => Iout=white (1)"
    2     "Ix~=zero | Iy~=zero => Iout=black (1)"
```

Matlab programme for Method C applied on Fig. b:

```
close all;
clear all;

I1 = imread('25_u.jpg');
figure, imshow(I1)
title('Original Image');
% text(size(I,2),size(I,1)+15, ...
%     'Image courtesy of Alan Partin', ...
%     'FontSize',7,'HorizontalAlignment','right');
% text(size(I,2),size(I,1)+25, ....
%     'Johns Hopkins University', ...
%     'FontSize',7,'HorizontalAlignment','right');
I=rgb2gray(I1);
[~,threshold] = edge(I,'sobel');
figure, imshow(I)
title('Gray Image')
fudgeFactor = 0.5;
BWs = edge(I,'sobel',threshold * fudgeFactor);

figure, imshow(BWs)
title('Binary Gradient Mask')

se90 = strel('line',3,90);
se0 = strel('line',3,0);

BWsdil = imdilate(BWs,[se90 se0]);
figure, imshow(BWsdil)
title('Dilated Gradient Mask')

sigma_i=0.01;
sigma_s=1;
[filtered, masked] = myBilateralFiltering(BWsdil, sigma_i, sigma_s);
figure, imshow(filtered)
title('Bilateral Filtered Image')
```

Matlab programme for Method D applied on Fig. 'b':

```
close all;
clear all;
I1 = imread('25_u.jpg');

figure, imshow(I1)
title('Original Image')

I=rgb2gray(I1);
figure, imshow(I)
title('Gray Image')


BW1 = edge(I,'sobel', 0.15);
figure, imshow(BW1)
title('Sobel')

BW2 = edge(I,'canny',[0.1 0.4]);
figure, imshow(BW2)
title('Canny')

BW3 = edge(I,'prewitt', 0.15);
figure, imshow(BW3)
title('Prewitt')

BW4 = edge(I,'roberts', 0.15);
figure, imshow(BW4)
title('Roberts')

BW5 = edge(I,'log', 0.01);
figure, imshow(BW5)
title('LoG')

BW6 = edge(I,'zerocross',0.01);
figure, imshow(BW6)
title('Zerocross')
```

Matlab programme for Method A applied on Fig. c:

```
%%load image
I=imread('26_u.jpg');
figure,imshow(I)
title('Original image')
%%Image adjust
Istrech = imadjust(I,stretchlim(I));
figure,imshow(Istrech)
title('Contrast stretched image')
%%Convert RGB image to gray
Igray_h = rgb2gray(Istrech);
figure,imshow(Igray_h,[])
title('RGB to gray (contrast stretched) ')
%%Image segmentation by thresholding
%use incremental value to run this selection till required threshold 'level' is
%achieved
level = 0.08;
gradmag = im2bw(Igray_h,level);
figure,imshow(gradmag)
title('Segmented cracks')
%%Image morphological operation
BW = bwmorph(gradmag,'clean',10);
figure,imshow(BW)
title('Cleaned image')
BW1 = bwmorph(gradmag,'thin', inf);
figure,imshow(BW1)
title('Thinned image')

% %%Image tool
% figure,imtool(BW1)
% figure,imtool(I)
%%Calaculate crack length
calibration_length=0.001;
calibration_pixels=1000;
crack_pixel=35;
crack_length=(crack_pixel *calibration_length)/calibration_pixels
```

```
crack_length =


   3.5000e-05
```

Matlab programme for Method B applied on Fig. b:

```
clear all;
close all;
Irgb = imread('26_u.jpg');
figure
image(Irgb,'CDataMapping','scaled')
title('Original Image')
Igray = rgb2gray(Irgb);

figure
image(Igray,'CDataMapping','scaled')
colormap('gray')
title('Input Image in Grayscale')
I = im2double(Igray);
Gx = [-1 1];
Gy = Gx';
Ix = conv2(I,Gx,'same');
Iy = conv2(I,Gy,'same');
figure
image(Ix,'CDataMapping','scaled')
colormap('gray')
title('Ix')
figure
image(Iy,'CDataMapping','scaled')
colormap('gray')
title('Iy')
edgeFIS = mamfis('Name','edgeDetection');
edgeFIS = addInput(edgeFIS,[-1 1],'Name','Ix');
edgeFIS = addInput(edgeFIS,[-1 1],'Name','Iy');
sx = 0.1;
sy = 0.1;
edgeFIS = addMF(edgeFIS,'Ix','gaussmf',[sx 0],'Name','zero');
edgeFIS = addMF(edgeFIS,'Iy','gaussmf',[sy 0],'Name','zero');
edgeFIS = addOutput(edgeFIS,[0 1],'Name','Iout');
wa = 0.1;
wb = 0.5;
wc = 1;
ba = 0;
bb = 0.5;
bc = 1;
edgeFIS = addMF(edgeFIS,'Iout','trimf',[wa wb wc],'Name','white');
edgeFIS = addMF(edgeFIS,'Iout','trimf',[ba bb bc],'Name','black');
figure
subplot(2,2,1)
plotmf(edgeFIS,'input',1)
title('Ix')
subplot(2,2,2)
plotmf(edgeFIS,'input',2)
title('Iy')
subplot(2,2,[3 4])
plotmf(edgeFIS,'output',1)
title('Iout')
r1 = "If Ix is zero and Iy is zero then Iout is white";
r2 = "If Ix is not zero or Iy is not zero then Iout is black";
edgeFIS = addRule(edgeFIS,[r1 r2]);
edgeFIS.Rules
Ieval = zeros(size(I));
for ii = 1:size(I,1)
    Ieval(ii,:) = evalfis(edgeFIS,[(Ix(ii,:));(Iy(ii,:))]');
end

figure
image(Ieval,'CDataMapping','scaled')
```

```
colormap('gray')
title('Edge Detection Using Fuzzy Logic')
```

```
ans =


  1×2 fisrule array with properties:


    Description

    Antecedent

    Consequent

    Weight

    Connection


  Details:

                        Description

          ____________________________________________


    1     "Ix==zero & Iy==zero => Iout=white (1)"

    2     "Ix~=zero | Iy~=zero => Iout=black (1)"
```

Published with MATLAB® R2020b

Matlab programme for Method C applied on Fig. c:

```
close all;
clear all;

I1 = imread('26_u.jpg');
figure, imshow(I1)
title('Original Image');
% text(size(I,2),size(I,1)+15, ...
%     'Image courtesy of Alan Partin', ...
%     'FontSize',7,'HorizontalAlignment','right');
% text(size(I,2),size(I,1)+25, ....
%     'Johns Hopkins University', ...
%     'FontSize',7,'HorizontalAlignment','right');
I=rgb2gray(I1);
[~,threshold] = edge(I,'sobel');
figure, imshow(I)
title('Gray Image')
fudgeFactor = 0.5;
BWs = edge(I,'sobel',threshold * fudgeFactor);

figure, imshow(BWs)
title('Binary Gradient Mask')

se90 = strel('line',3,90);
se0 = strel('line',3,0);

BWsdil = imdilate(BWs,[se90 se0]);
figure, imshow(BWsdil)
title('Dilated Gradient Mask')

sigma_i=0.01;
sigma_s=1;
[filtered, masked] = myBilateralFiltering(BWsdil, sigma_i, sigma_s);
figure, imshow(filtered)
title('Bilateral Filtered Image')
```

Matlab programme for Method D applied on Fig. c:

```
close all;
clear all;
I1 = imread('26_u.jpg');

figure, imshow(I1)
title('Original Image')

I=rgb2gray(I1);
figure, imshow(I)
title('Gray Image')


BW1 = edge(I,'sobel', 0.15);
figure, imshow(BW1)
title('Sobel')

BW2 = edge(I,'canny',[0.1 0.4]);
figure, imshow(BW2)
title('Canny')

BW3 = edge(I,'prewitt', 0.15);
figure, imshow(BW3)
title('Prewitt')

BW4 = edge(I,'roberts', 0.15);
figure, imshow(BW4)
title('Roberts')

BW5 = edge(I,'log', 0.01);
figure, imshow(BW5)
title('LoG')

BW6 = edge(I,'zerocross',0.01);
figure, imshow(BW6)
title('Zerocross')
```