%-- 19/12/2020 15:45 --%

HW4\_Q2a

HW4\_Q2aa

order\_reduction

HW4\_Q2aa

HW4\_Q1

%-- 20/12/2020 19:37 --%

acos(-0.5)

rad2deg(acos(-0.5))

rad2deg(acos(0.5))

HW4\_Q4

rad2deg(y(end,1))

acos(0.5)

acos(-0.5/-1)

acos(-0.5/1)

rad2deg(ans)

acos(-0.5/-1)

rad2deg(ans)

HW4\_Q4

rad2deg(y(end,1))

rad2deg(y(end,2))

%-- 21/12/2020 10:54 --%

phi=32+46/60+44.34472/3600;

lambda=35+1/60+22.74061/3600;

h=225.046;

IGD12toIG12( phi, lambda, h )

rad2deg(lambda)

digits 8

rad2deg(lambda)

digits 10

rad2deg(lambda)

phi=32+46/60+44.34472/3600;

lambda=35+1/60+22.74061/3600;

h=225.046;

IGD12toIG12( phi, lambda, h )

rad2deg(lambda)

format long

rad2deg(lambda)

X

format long

X

rad2deg(phi)

(ans-32-46/60)\*3600

ans

h

%-- 21/12/2020 12:28 --%

phi=32+46/60+44.34472/3600;

lambda=35+1/60+22.74061/3600;

h=225.046;

IGD12toIG12( phi, lambda, h )

h

[E N]=IGD12toIG12( phi, lambda, h )

E

%-- 21/12/2020 13:51 --%

order\_reduction

HW4\_Q5

%-- 24/12/2020 13:30 --%

pdetool

pderect

pderect([10 20 10 20])

%-- 07/01/2021 16:39 --%

explicit\_full

PDE\_Parabolic\_ADI

%-- 05/06/2021 10:57 --%

0.78\*10/7

67-1.11\*63

67-1.14\*63

67-1.11\*63

56+2.93

/1.11

ans/1.11

77-1.11\*63

0.78\*sqrt(23)/sqrt(1-0.78^2)

1.11-2.069\*1.11/5.978

1.11+2.069\*1.11/5.978

8250\*20

0.47\*sqrt(4090)

7051/20

(6430-8250)/sqrt(352.55)

-0.87\*5.297/1.245

47.276+3.701\*10.328

47.276+3.701518\*10.328

0.87^2

100-75.7

sqrt((0.87^2)\*30)/sqrt(1-0.87^2)

-3.701-2.042\*(-3.701/-9.665)

-3.701+2.042\*(-3.701/-9.665)

A=[1 721 1.1

1 721 0.62

1 721 0.31

1 810 1.1

1 810 0.62

1 810 0.31

1 975 1.1

1 975 0.62

1 975 0.31

1 1246 1.1

1 1246 0.62

1 1246 0.31

];

l=0.235

0.105

0.057

0.126

0.066

0.032

1.007

0.554

0.323

2.941

1.625

0.944

l=[0.235

0.105

0.057

0.126

0.066

0.032

1.007

0.554

0.323

2.941

1.625

0.944

];

b=inv(A'\*A)\*(A'\*l);

b

SSE=l'\*l-b'\*A'\*l

sqrt(SSE/8)

1-1.524/8.344

1-ans

1-(11/7)\*ans

A=[1 56 10 560

1 56 10 560

1 56 20 1120

1 56 20 1120

1 76 10 760

1 76 10 760

1 76 20 1520

1 56 20 1120

];

l=[23.5

25

28.2

28.3

22.3

20.7

16.8

15.7

];

b=inv(A'\*A)\*(A'\*l);

b

sqrt(1.524/9)

1-(11/8)\*0.183

1-(11/9)\*0.183

A=[1 721 1.1

1 721 0.62

1 721 0.31

1 810 1.1

1 810 0.62

1 810 0.31

1 975 1.1

1 975 0.62

1 975 0.31

1 1246 1.1

1 1246 0.62

1 1246 0.31

];

N=inv(A'\*A)

diag(N)

sqrt(N)

sqrt(diag(N))

nn=sqrt(diag(N))

N=inv(A'\*A)

diag(N)

nn=sqrt(ans)

0.003-2.262\*0.411\*nn(2)

0.003+2.262\*0.411\*nn(2)

0.942-2.262\*0.411\*nn(3)

0.942+2.262\*0.411\*nn(3)

A=[1 56 10 560

1 56 10 560

1 56 20 1120

1 56 20 1120

1 76 10 760

1 76 10 760

1 76 20 1520

1 56 20 1120

];

l=[23.5

25

28.2

28.3

22.3

20.7

16.8

15.7

];

N=inv(A'\*A);

nn=diag(N)

b=inv(A'\*A)\*(A'\*l);

b

SSE=l'\*l-b'\*A'\*l

sqrt(SSE/4)

b

S=ans

b(2)/(S\*sqrt(nn(2)))

nn=sqrt(nn)

b(2)/(S\*(nn(2)))

b

b(3)/(S\*(nn(3)))

b(4)/(S\*(nn(4)))

b

1-0.005/2

-0.258\*5.34/10.35

-0.258\*sqrt(6)/sqrt(1-0.258^2)

1-0.005/2

x=[609,594,581];

y=[664,621,638];

inv(x'\*x)\*(x'\*y)

A=[1 664;

1 621;

1 638];

l=[609;594;581];

b=inv(A'\*A)\*(A'\*l);

b

AA=[1 l(1);1 l(2) 1 l(3)];

AA=[1 l(1);1 l(2); 1 l(3)];

ll = A(:,2);

bb=inv(AA'\*AA)\*(AA'\*ll)

A=AA

l=ll

b=inv(AA'\*AA)\*(AA'\*ll)

sqrt(2.22^2/8+1.90^2/8)

(5.75-7.88)/ans

s1=2.22;

s2=1.90;

(s1^2/8+s2^2/8)^2

(1/7)\*(s1^2/8)^2

(1/7)\*(s2^2/8)^2

1.13912929

ans/(0.0542168003571429+0.0290895089285714)

(0.0542168003571429+0.0290895089285714)

1.139/ans

52.5-0.5\*8\*9

83.5-0.5\*8\*9

5.4\*2.355

11.78/2.355

%-- 15/12/2021 09:02 --%

hw1

RGB = [R(row,col);G(row,col);B(row,col)];

R(row(1),col(1))

hw1

round(col)

hw1

%-- 21/12/2021 11:00 --%

hw1

%-- 28/12/2021 09:27 --%

hw1b

rand()

rand(2,2)

rand(2,2,2)

%-- 28/12/2021 11:03 --%

rand(1,4)

randi()

randi(4)

randi(100)

hw1

A=rand(3);

A(1,3)

Dist = NaN(m,n);

Dist = NaN(10,1);

min(dist)

min(Dist)

hw1b

hw1

R(groupsXY(1,:))

groupsXY(1,:)

R[ans]

R(ans)

R(135,143)

R(1494,1790)

hw1

hw1b

K = zeros(5,4){3}

hw1b

hw1

K=R(centroidsXY);

K=R(centroidsXY');

R(centroidsY,centroidsX)

R([centroidsY,centroidsX])

R(centroidsY(1),centroidsX(1)])

R(centroidsY(1),centroidsX(1))

R(centroidsY,centroidsX)

R([centroidsY],[centroidsX])

hw1b

A(1:5,1:5,1)

size(A,1)

A(1:5,1:4,1)

hw1

A = rand(3,3,3);

[YY,ClusterMapY] = min(A,[],3);

YY

ClusterMapY

hw1

hw1b

hw1

M=R(clusterResults(:)==i);

hw1

std(R(clusterResults(:)==i))

hw1

L = R(clusterResults(:)==1);

D = L-clustersRGB\_new(1,1);

D = abs(D);

max(D)

hw1

acos

acos(5)

acos(1)

hw1

T = acos(a./(b.\*c));

T = ValuesMat(:,:,1);

T = ValuesMat(:,:,2);

T = ValuesMat(:,:,3);

T = ValuesMat(:,:,4);

T = ValuesMat(:,:,5);

T = ValuesMat(:,:,6);

T = ValuesMat(:,:,3);

Z = imag(T);

T = ValuesMat(:,:,2);

T(25,31)

T = ValuesMat(:,:,3);

i=3;

a = RGB(:,:,1).\*clustersRGB(i,1)+RGB(:,:,2).\*clustersRGB(i,2)+RGB(:,:,3).\*clustersRGB(i,3);

b = sqrt(RGB(:,:,1).^2+RGB(:,:,2).^2+RGB(:,:,3).^2);

c = sqrt(clustersRGB(i,1).^2+clustersRGB(i,2).^2+clustersRGB(i,3).^2);

aa=a(36,38)

bb=b(36,38)

cc=c(36,38)

cc

cc=c

aa/(bb\*cc)

acos(1)

TT = acos((a./b).\*(a./c));

TT(36,38)

acos(a./(b.\*c));

TT = round(acos(a./(b.\*c)),8);

TT = acos(round(a./(b.\*c),8));

TT(36,38)

TT = double(acos(a./(b.\*c));

TT = double(acos(a./(b.\*c)));

TT(36,38)

TT = acos(round(a./(b.\*c),10));

TT = acos(round(a./(b.\*c),11));

TT = acos(round(a./(b.\*c),16));

TT = acos(round(a./(b.\*c),14));

hw1

d(36,38)

hw1

max(d)

max(max(d))

V = imag(ValuesMat(:,:,3))

d(25,5)

hw1

%-- 04/01/2022 10:09 --%

hw1

C = cov(R\_class,G\_class,B\_class);

C = cov([R\_class,G\_class,B\_class]);

C = cov([R\_class;G\_class;B\_class]);

hw1

-0.5\*(RGB(j,k,i)-clustersRGB(i,:))'\*inv(Sk)\*(RGB(j,k,i)-clustersRGB(i,:))

(RGB(j,k,i)-clustersRGB(i,:))'

hw1

max(ValuesMat(:,:,2))

hw1

max(ValuesMat(:,:,1))

hw1

clustersRGB(i,:)

RGB(j,k,:)

hw1

RGB(1,1,:)

RGB(1,1,:)-clustersRGB(1,:)

hw1

-0.5\*([R(j,k),G(j,k),B(j,k)]-clustersRGB(i,:))/(Sk)

(Sk)\([R(j,k),G(j,k),B(j,k)]-clustersRGB(i,:))'

([R(j,k),G(j,k),B(j,k)]-clustersRGB(i,:))/(Sk)\*([R(j,k),G(j,k),B(j,k)]-clustersRGB(i,:))'

hw1

ValuesMat(1,1,1)

ValuesMat(100,100,1)

x = [RGB(100,100,:)]

y=clustersRGB(1,:)

x = RGB(100,100,:)

x = [RGB(100,100,1),RGB(100,100,2),RGB(100,100,3)];

(x-y)\*(x-y)'

sqrt(ans)

hw1\_clean

RGB(:,:,1:3)-clustersRGB(i,:)

hw1\_clean

norm(clustersRGB\_new-clustersRGB,inf)

clustersRGB\_new

hw1\_clean

norm(clustersRGB\_new-clustersRGB,inf)

hw1\_clean

untitled

distinct(initClustMap)

unique(initClustMap)

untitled

unique(initClustMap)

untitled

unique(initClustMap)

untitled

A = rand(5,1)

[M,I] = min(A)

A = [1 2 3 5];

norm(A)

1^2+2^2+3^2+5^2

sqrt(ans)

%-- 05/01/2022 12:20 --%

hw1\_minimum\_distance

norm(clustersRGB\_std(k,:))

hw1\_minimum\_distance

norm(clustersRGB\_std(index,:))

hw1\_minimum\_distance

clustersRGB\_new-clustersRGB

norm(clustersRGB\_new-clustersRGB)

hw1\_minimum\_distance

norm(clustersRGB\_new-clustersRGB)

A=2;

std(A)

hw1\_minimum\_distance

map = [clustersRGB/255];

clusterResults(clustersResults==0)=nClusters;

subplot(122),imagesc(clusterResults);title('Clusters');colormap(map);

map = [clustersRGB/255];

clusterResults(clusterResults==0)=nClusters;

subplot(122),imagesc(clusterResults);title('Clusters');colormap(map);

clustersRGB(:,1) = 0:floor(255/(nClusters-1)):255

clustersRGB(:,1) = 0:floor(255/(nClusters)):255

hw1\_minimum\_distance

length(clusterResults==11)

size(clusterResults==11)

hw1\_minimum\_distance

unique(clusterResults)

hw1\_minimum\_distance

A=[1 1;2 2];

iszero(A)

A(A==0)

A(A==1)

length(A(A==0))

hw1\_minimum\_distance

tempStd = norm(std(R(clusterResults(:)==index)),std(G(clusterResults(:)==index)),std(B(clusterResults(:)==index)))

norm([std(R(clusterResults(:)==index)),std(G(clusterResults(:)==index)),std(B(clusterResults(:)==index])))

norm([std(R(clusterResults(:)==index)),std(G(clusterResults(:)==index)),std(B(clusterResults(:)==index))])

norm([std(R(clusterResults(:)==index-1)),std(G(clusterResults(:)==index-1)),std(B(clusterResults(:)==index-1))])

norm([std(R(clusterResults(:)==index+1)),std(G(clusterResults(:)==index+1)),std(B(clusterResults(:)==index+1))])

hw1\_minimum\_distance

norm(clustersRGB\_new-clustersRGB)

hw1\_minimum\_distance

norm(clustersRGB\_new-clustersRGB)

map = [clustersRGB/255];

subplot(122),imagesc(clusterResults);title('Clusters');colormap(map);

sum(sum(clusterResults==11))

hw1\_minimum\_distance

std([187 192 188 185])

max([187 192 188 185])-min([187 192 188 185])

mean([187 192 188 185])

hw1\_minimum\_distance

isempty(clusterResults(clusterResults==0))

sum(sum(clusterResults==0))

m\*n

hw1\_minimum\_distance

sum(sum(clusterResults==0))

hw1\_minimum\_distance

sum(sum(clusterResults==0))

hw1\_minimum\_distance

sum(sum(clusterResults==0))

unique(clusterResults)

map = [0 0 0;clustersRGB/255];

subplot(122),imagesc(clusterResults);title('Clusters');colormap(map);

hw1\_minimum\_distance

sum(sum(clusterResults==0))

hw1\_minimum\_distance

unique(initClustMap)

hw1\_minimum\_distance

hw1

hw1\_minimum\_distance

%-- 05/01/2022 18:03 --%

hw1\_minimum\_distance

%-- 05/01/2022 18:15 --%

hw1\_minimum\_distance

isempty(k==27)

isempty(k)

isempty(k==27)

sum(k==27)

sum(k==28)

hw1\_minimum\_distance

sum(k==27)

hw1\_minimum\_distance

K = img(col{i},row{i},:);

K = img(col{1},row{1},:);

col{1}

K = img([col{1},row{1},:]);

K = img([col{1},row{1}],:);

hw1

hw1\_minimum\_distance

img(temp1,temp2,:)

img(temp1',temp2',:)

RGB(temp1',temp2',:)

RGB(temp1,temp2,:)

R(temp1,remp2)

R(temp1,temp2)

R([temp1,temp2])

R(1,1)

R([1,2],[1,2])

img([temp1],[temp2]:););

end

img([temp1],[temp2],:)

R([temp1],[temp2])

R([temp1]',[temp2]')

R(sub2ind(size(R), [2 3 4], [1 2 4]))

R(sub2ind(size(R), temp1, temp2))

R(sub2ind(size(R), [temp1], [temp2]))

temp1 = round(col{i});

temp2 = round(row{i});

R(sub2ind(size(R), temp1, temp2))

img([temp1],[temp2],:)

R([temp1],[temp2])

R(temp1,temp2)

R(sub2ind(size(R), temp1, temp2))

hw1\_minimum\_distance

R(sub2ind(size(R), temp1, temp2))

hw1\_minimum\_distance

R(sub2ind(size(R), temp1, temp2))

hw1\_minimum\_distance

%-- 09/01/2022 09:50 --%

hw1\_final

A=inf;

B=0;

B=inf;

min(A,B)

hw1\_final

sqrt(clustersRGB\_std(1,1)+clustersRGB\_std(1,2)+clustersRGB\_std(1,3))

sqrt(temp\_std)

sqrt(clustersRGB\_std(1,1)^2+clustersRGB\_std(1,2)^2+clustersRGB\_std(1,3)^2)

A = rand(4)

A = 10\*rand(4)

A(A>7)=inf

min(min(A))

min(4,NaN)

hw1\_final

A=ValuesMat\_SAM(:,:,1);

A=min(min(clusterResults\_MD))

A=ValuesMat\_MD(:,:,1);

B=ValuesMat\_MD(:,:,2);

C=ValuesMat\_MD(:,:,3);

clusterResults\_MD(isnan(minValues\_MD))=0;

hw1\_final

i=1;

temp\_col = round(col{i});

temp\_row = round(row{i});

A=[R(sub2ind(size(R), temp\_row, temp\_col))];

i=2;

temp\_col = round(col{i});

temp\_row = round(row{i});

A=[R(sub2ind(size(R), temp\_row, temp\_col))];

hw1\_final

i=1;

temp\_col = round(col{i});

temp\_row = round(row{i});

A=R(sub2ind(size(R), temp\_row, temp\_col))

R(temp\_col(1),temp\_row(1))

R(temp\_col(2),temp\_row(2))

R(temp\_col(3),temp\_row(3))

hw1\_final

A=R(sub2ind(size(R), temp\_row, temp\_col))

R(temp\_col(1),temp\_row(1))

R(temp\_row(1),temp\_col(1))

R(temp\_row(2),temp\_col(2))

R(temp\_row(3),temp\_col(3))

hw1\_final

norm(distR,distG,distB)

norm([distR,distG,distB])

hw1\_final

A = sqrt(distR(1)^2+distG(1)^2+distB(1)^2);

B = sqrt(distR(2)^2+distG(2)^2+distB(2)^2);

C = sqrt(distR(3)^2+distG(3)^2+distB(3)^2);

D = sqrt(distR(4)^2+distG(4)^2+distB(4)^2);

std = ([A B C D])

std = ([A B C D]')

std = (A,B,C,D)

M = [A,B,C,D];

std = std(M)

std = std(M')

std = std([A B C D])

std = std(A B C D)

std = std([A; B; C; D])

std = ([A B C D])

temp = ([A B C D])

std(temp)

V = [A,B,C,D];

K = std(A)

K = std(V)

std(clustersRGB\_std(1,:)

std(clustersRGB\_std(1,:))

sqrt(distR.^2+distG.^2+distB.^2)

std(sqrt(distR.^2+distG.^2+distB.^2)

std(sqrt(distR.^2+distG.^2+distB.^2)))

std(sqrt(distR.^2+distG.^2+distB.^2))

hw1\_final

acos(round(tempVal,12))

A=acos(round(tempVal,12))

deg(A)

rad2deg(A)

hw1\_final

tempV = -0.5\*([sampled\_R{i},sampled\_G{i},sampled\_B{i}]-clustersRGB(i,:))\*((Sk)\([sampled\_R{i},sampled\_G{i},sampled\_B{i}]-clustersRGB(i,:))');

f\_k = (1/(((2\*pi)^(3/2))\*sqrt(det(Sk))))\*exp(tempV);

A=-log(f\_k)-0.5\*log(det(Sk))+(tempV);

std2(A)

hw1\_final

sum(sum(isnan(maxValues\_ML)))

unique(clusterResults\_ML)

hw1\_final

ValuesMat\_ML2 = zeros(m,n,nClusters); % Maximum Likelhood values

A = [tempR,tempG,tempB]-clustersRGB(i,:);

C = A\*inv(Sk)\*A';

hw1\_final

tempV = -0.5\*([tempR,tempG,tempB]-clustersRGB(i,:))\*((Sk)\([tempR,tempG,tempB]-clustersRGB(i,:))');

f\_k = (1/(((2\*pi)^(3/2))\*sqrt(det(Sk)))).\*exp(tempV);

result = -log(f\_k)-0.5\*log(det(Sk))+(tempV);

A=ValuesMat\_ML(:,:,1)-result;

A=-0.5\*([tempR,tempG,tempB]-clustersRGB(i,:));

B = A.\*inv(Sk);

B = A\*inv(Sk);

C=([tempR,tempG,tempB]-clustersRGB(i,:))';

C=([tempR,tempG,tempB]-clustersRGB(i,:));

C=([tempR,tempG,tempB]-clustersRGB(i,:))';

A = B\*C;

f\_k = (1/(((2\*pi)^(3/2))\*sqrt(det(Sk))))\*exp(A);

result = -log(f\_k)-0.5\*log(det(Sk))+(A);

D = diag(result);

M = reshape (D,m,n);

V = M-ValuesMat\_ML(:,:,1);

max(V)

max(ValuesMat\_ML(:,:,1))

[maxValues\_ML,clusterResults\_ML] = max(ValuesMat\_ML,[],3);

hw1\_final

sum(sum(maxValues\_ML==0))

sum(sum(isnan(maxValues\_ML)))

unique(clusterResults\_ML)

hw1\_final

unique(clusterResults\_ML)

subplot(224),imagesc(clusterResults\_ML);title('Maximum Likelhood');colormap(map);

map1 = [0 0 0;clustersRGB/255];

subplot(224),imagesc(clusterResults\_ML);title('Maximum Likelhood');colormap(map1);

hw1\_final

clusterResults\_ML(40,160)

maxValues\_ML(40,160)

maxValues\_ML(60,80)

maxValues\_ML(140,200)

max(max(maxValues\_ML))

hw1\_final

%-- 10/01/2022 13:52 --%

untitled

hw1\_final

%-- 10/01/2022 14:44 --%

hw1\_final

max(max(maxValues\_ML))

hw1\_final

max(max(maxValues\_ML))

%-- 11/01/2022 09:09 --%

hw1\_final

NO\_CLUSTER = [ 1 1 1];

%-- 18/01/2022 16:20 --%

hw1\_final

wavelet

W = file.h;

wavelet

hw1\_final

wavelet

hw1\_final

wavelet

hw1\_final

%-- 18/01/2022 17:40 --%

hw1\_final

wavelet

hw1\_final

%-- 18/01/2022 17:52 --%

hw1\_final

imsave

hw1\_final

wavelet

%-- 19/03/2022 17:57 --%

wavelet

%-- 22/03/2022 08:43 --%

clc

tut01

%-- 04/04/2022 16:03 --%

new\_wavelet\_2

%-- 04/04/2022 16:51 --%

W = [0 0 0 0 100 100 100 100 0 0 100 100 0 100 100 0 50];

W = W(1:16)

new\_wavelet\_2

file=file(1:1024);

file=file(1:1024,:);

file=file(1:1024,1);

load('W3.mat')

file = {file};

new\_wavelet\_2

W = W(1:1024);

C = [sqrt(2) sqrt(2)];

W=W(1:256);

%-- 08/04/2022 21:38 --%

hw1

hdr1 = envihdrread('VE\_VM01\_VSC\_PDTIMG\_L1VALD\_ISRAES02\_20190318.DBL\_subset.hdr');

hdr1

hdr2 = envihdrread('VE\_VM01\_VSC\_PDTIMG\_L2VALD\_ISRAES02\_20190318\_FRE.DBL\_subset.hdr');

wavelengths1 = strtrim(split(string(hdr1.wavelength),','));

wavelengths2 = strtrim(split(string(hdr2.wavelength),','));

tut01

hw1

imshow(Im2)

imshow(uint16(Im2))

hw1

%-- 10/04/2022 13:09 --%

new\_wavelet

%-- 11/04/2022 11:42 --%

hw2

K = roi{1}

hw2

roi{1}.Position

roi{2}.Position

max(max(Reflectance\_L2))

max(max(IM\_imadjust2(:,:,1)))

hw2

K.Polygon = roi{i};

K.Position

K.Polygon.Position

hw2

K.Polygon = roi{1};

K

K = Polygon()

K=Polygon.empty

hw2

K=Reflectance\_L2(:,:,5);

S=K>1;

sum(S)

sum(sum(S))

K=Reflectance\_L2(:,:,4);

S=K>1;

sum(sum(S))

uint8(1000)

uint8(-1000)

hw2

A = in.\*double(IM\_imadjust1(:,:,1));

count(A>0)

max(max(A))

sum(sum(in))

6778\*3047

A = IM\_imadjust1(:,:,1);

B = IM\_imadjust1(in);

mean(B)

std(B)

std(double(B))

mean(double(B))

hw2

max(max(A))

min(min(A))

hw2

load('poly1.mat')

save('poly1.mat','poly')

load('poly1.mat')

load('poly1.mat','poly1')

load('poly1.mat')

clc

clear

load('poly1.mat')

load('poly1.mat','p')

load('poly1.mat',p)

hw2

roi = images.roi.Polygon;

draw(roi);

poly = roi.Position;

xv=roi.Position(:,1);

yv=roi.Position(:,2);

[xq, yq]= meshgrid (1:3047,1:6778 );

in = inpolygon(xq,yq,xv,yv);

sum(sum(in))

figure,imshow(in)

A = double(IM\_imadjust1(:,:,1));

B = double(IM\_imadjust1(:,:,2));

C = double(IM\_imadjust1(:,:,3));

AA = A(in);

BB=B(in);

CC=C(in);

mean(AA)

std(AA)

mean(BB)

std(BB)

mean(CC)

std(CC)

332/56000

hw2

%-- 11/04/2022 17:28 --%

hw2

load('polygons.mat')

stats1.numPixels{1} = sum(sum(in1));

stats1.numPixels(1) = sum(sum(in1));

sum(sum(in1))

%-- 11/04/2022 19:02 --%

load('stats1.mat')

%-- 12/04/2022 10:36 --%

load('polygons.mat')

poly{9}=[]

poly{9}={}

poly=poly{1:8};

load('polygons.mat')

temp = poly{7};

poly{7}=poly{8};

poly{8}=temp;

hw2

load('polygons.mat')

xv=temp(:,1);

yv=temp(:,2);

xv=temp(:,1)

K=xv/2;

L=xv\*m2/m1;

T=K-L;

T

m2/m1

n2/n1

stats1.Properties.VariableNames(3)

stats1.Properties.VariableNames{3}

t1 = [t1,table(zeros(numOfRois,1))];

t1.Properties.VariableNames{2+i}=strcat('band\_',string(i));

t1.Properties.VariableNames{2+i}='ggg';

strcat('band\_',string(i))

char(ans)

t1.Properties.VariableNames{2+i}=char(strcat('band\_',string(i)));

avgs1(1,1)

stats1(1,1)

stats1{1,1}

stats1{1,2}

%-- 12/04/2022 16:44 --%

hw2

i=1;

polyfit(avgs1{:,2+i},avgs2{:,2+i},1)

roiPositions = roiPositions';

hw2

C = get (gca, 'CurrentColor');

Calibrated\_L1 = zeros(nColumns1,nRows1,length(coeffs));

A=imadjust(DN\_image(:,:,1));

A=imadjust(DNImage(:,:,1));

Calibrated\_L1 = zeros(nColumns1,nRows1,length(coeffs));

Calibrated\_L1(:,:,i) = imadjust(DN\_image(:,:,i))\*tempCoeff(1)+tempCoeff(2);

Calibrated\_L1(:,:,1) = imadjust(DNImage(:,:,1))\*tempCoeff(1)+tempCoeff(2);

tempCoeff = coeffs{1};

Calibrated\_L1(:,:,1) = imadjust(DNImage(:,:,1))\*tempCoeff(1)+tempCoeff(2);

hw2

max(max(Calibrated\_L1(:,:,1)))

max(max(Calibrated\_L1(:,:,2)))

max(max(Calibrated\_L1(:,:,3)))

max(max(Calibrated\_L1(:,:,4)))

%-- 12/04/2022 18:23 --%

hw1\_final

DN\_image = imread('VE\_VM01\_VSC\_PDTIMG\_L1VALD\_ISRAES02\_20190318.DBL\_subset.tif');

band{1}=imadjust(DN\_image(:,:,5));

band{2}=imadjust(DN\_image(:,:,4));

band{3}=imadjust(DN\_image(:,:,3));

IM\_imadjust1=cat(3, band{3},band{2},band{1});

[nRows1,nColumns1,nBands1] = size(DN\_image);

hw2

close

hw2

%-- 12/04/2022 20:27 --%

hw2

A=double(imadjust(DN\_image)).\*ones(15,1)

sum(keys>3)

sum(keys1>3)

sum(keys2>3)

hw2

%-- 29/04/2022 18:55 --%

hw3\_main

dirFiles.name

K=dirFiles.name

K=dirFiles(3).name

hw3\_main

max(tempY)

sum(tempY>1)

sum(tempY>100)

hw3\_main

mean([1,2,3])

mean([1,2,3,NaN])

nanmean([1,2,3,NaN])

hw3\_main

K=[1 2 2 3;1 2 3 4];

K=K';

unique(K(:,1))

[a b]=unique(K(:,1));

K(b,:)

K2=ans;

K2(K2(:,2)>3)=3

A = [1 2 3 4 50 6 7];

B = [10,11,12,13,14,15,16]';

A=A';

A(A<3)=[];

A = [1 2 3 4 50 6 7]';

B(A<3)=[];

hw3\_main

K=files{14};

A=K{:,1};

for i=1:2151

B(i,1) = A(i+1)-A(i);

end

max(B)

C = [1,2,4,3,5];

[s,t]=sort(C);

s

t

C = [10,20,40,30,50];

[s,t]=sort(C);

t

C(t)

hw3\_main

K=files{14};

A=K{:,1};

for i=1:2151

B(i,1) = A(i+1)-A(i);

end

max(B)

min(B)

A=wavelengths{14};

for i=1:2151

B(i,1) = A(i+1)-A(i);

end

for i=1:2139

B(i,1) = A(i+1)-A(i);

end

max(B)

min(B)

hw3\_main

newWL = [MIN\_WL:MAX\_WL];

MAX\_WL-MIN\_WL

newRefl = NaN(MAX\_WL-MIN\_WL+1,1);

hw3\_main

sum(isnan(newRefl))

hw3\_main

sum(isnan(tempWL))

find(isnan(tempWL))

hw3\_main

A=(2:5)

A=[2:5]

A=(2:5)'

hw3\_main

size(MATERIALS\_INTERVALS,1)

A=[ones(5,1);2\*ones(5,1)];

A=[ones(5,1),2\*ones(5,1)];

A=[ones(5,1),2\*ones(5,1),3\*ones(5,1),ones(5,1)];

mean(A(:,2:3))

mean(A(:,2:3)')

hw3\_main

sum(isnan(oldRefl))

sum(isnan(oldRefl>100))

temp = oldRefl;

K=oldRefl-temp;

max(K)

min(K)

sum(isnan(K))

sum(temp>100)

max(temp)

temp = oldRefl;

max(temp)

sum(temp>110)

hw3\_main

clear all

hw3\_main

samplings{1} = [col1{1};row1{1}]

[col1{1};row1{1}]

[col1{1},row1{1}]

samplings{1} = [col1{1},row1{1}]

samplings{2} = [col1{2},row1{2}]

samplings{3} = [col1{3},row1{3}]

samplings{4} = [col2{1},row2{1}]

samplings{5} = [col2{2},row2{2}]

samplings{6} = [col2{3},row2{3}]

samplings{7} = [col2{4},row2{4}]

samplings{8} = [col2{5},row2{5}]

samplings=samplings'

load('samplings.mat')

A = hcube1.Wavelength;

B = hcube2.Wavelength;

sum(A-B)

hw3\_main

sum(sum(M>1))

clear all

col1=col1';

row1=row1';

col2=col2';

row2=row2';

commandhistory

samplings{1} = [col1{1},row1{1}]

samplings{2} = [col1{2},row1{2}]

samplings{3} = [col2{1},row2{1}]

samplings{4} = [col1{3},row1{3}]

samplings{5} = [col2{2},row2{2}]

samplings{6} = [col2{3},row2{3}]

samplings{7} = [col2{4},row2{4}]

samplings{8} = [col2{5},row2{5}]

samplings=samplings';

15\*7+20

hw3\_main

MATERIALS\_INTERVALS\_HSC(1,1):MATERIALS\_INTERVALS\_HSC(1:2)

MATERIALS\_INTERVALS\_HSC(1:2)

MATERIALS\_INTERVALS\_HSC(1:1)

hw3\_main

sum(sum(isnan(reflectanceFromHSC)))

sum(sum(isnan(reflectancesFromHSC)))

p{1}=samplings{1};

p{2}=samplings{2};

p{3}=samplings{4};

p{1}=samplings{3};

p{2}=samplings{5};

p{3}=samplings{6};

p{4}=samplings{7};

p{5}=samplings{8};

wavelengthsFromHSC(1200)

wavelengthsFromHSC(120)

wavelengthsFromHSC(130)

wavelengthsFromHSC(230)

wavelengthsFromHSC(170)

wavelengthsFromHSC(150)

wavelengthsFromHSC(170)

wavelengthsFromHSC(160)

wavelengthsFromHSC(155)

wavelengthsFromHSC(157)

indexesTable{1,3:end}=R;

f = f + e\_prc(:,i).^2;

e\_prc(:,i).^2;

e\_prc(:,i).^2

%-- 24/05/2022 09:11 --%

load('samplings.mat')

num=7

colors(7,:)=[0 0 0];

colors = [0 0.4470 0.7410;...

0.8500 0.3250 0.0980;...

0.9290 0.6940 0.1250;...

0.5940 0.2840 0.5560;...

0.4660 0.6740 0.1880;...

0.3010 0.7450 0.9330;...

0.2 0.2 0.2];

imshow(B==1)

K=(B==1);

imshow(K)

imshow(L==1)

imshow(L==2)

imshow(L==3)

imshow(L==4)

imshow(L==5)

imshow(L==6)

imshow(L==7)

imshow(L==8)

load('BW\_2.mat')

A=B(L);

B(:,:,1) = double(B(:,:,1)).\*imrotate(BW\_2,90);

B(:,:,2) = double(B(:,:,2)).\*imrotate(BW\_2,90);

B(:,:,3) = double(B(:,:,3)).\*imrotate(BW\_2,90);

imshow(B)

load('BW\_1.mat')

load('L3.mat')

imshow(L==1)

imshow(L==2)

imshow(L==3)

imshow(L==4)

imshow(L==5)

imshow(L==6)

imshow(L==7)

B(:,:,1) = double(B(:,:,1)).\*L(L==1);

A=L(L==1);

A=(L==1);

B(:,:,1) = double(B(:,:,1)).\*A;

B(:,:,2) = double(B(:,:,2)).\*A;

B(:,:,3) = double(B(:,:,3)).\*A;

imshow(B)

B = labeloverlay(rgbImg2,L);

A=(L~=1);

B(:,:,1) = double(B(:,:,1)).\*A;

B(:,:,2) = double(B(:,:,2)).\*A;

B(:,:,3) = double(B(:,:,3)).\*A;

imshow(B)

load('L1.mat')

load('L2.mat')

close all

%-- 27/05/2022 18:22 --%

load('samplings.mat')

%-- 29/05/2022 10:21 --%

app.nRows

app.nColumns

app.nBands

k = logic(3,1)

k = bool(3,1)

k = log(3,1)

A = rand(10,10)

B = (A>0.2);

C = logical(3,1);

C = logical(3);

hw5\_test

A(1,1)-B(1,1)

C=A-B;

max(max(C))

app.roiPositions{2}

%-- 30/05/2022 13:39 --%

wavelength = {

397.32,

400.20,

403.09,

405.97,

408.85,

411.74,

414.63,

417.52,

420.40,

423.29,

426.19,

429.08,

431.97,

434.87,

437.76,

440.66,

443.56,

446.45,

449.35,

452.25,

455.16,

458.06,

460.96,

463.87,

466.77,

469.68,

472.59,

475.50,

478.41,

481.32,

484.23,

487.14,

490.06,

492.97,

495.89,

498.80,

501.72,

504.64,

507.56,

510.48,

513.40,

516.33,

519.25,

522.18,

525.10,

528.03,

530.96,

533.89,

536.82,

539.75,

542.68,

545.62,

548.55,

551.49,

554.43,

557.36,

560.30,

563.24,

566.18,

569.12,

572.07,

575.01,

577.96,

580.90,

583.85,

586.80,

589.75,

592.70,

595.65,

598.60,

601.55,

604.51,

607.46,

610.42,

613.38,

616.34,

619.30,

622.26,

625.22,

628.18,

631.15,

634.11,

637.08,

640.04,

643.01,

645.98,

648.95,

651.92,

654.89,

657.87,

660.84,

663.81,

666.79,

669.77,

672.75,

675.73,

678.71,

681.69,

684.67,

687.65,

690.64,

693.62,

696.61,

699.60,

702.58,

705.57,

708.57,

711.56,

714.55,

717.54,

720.54,

723.53,

726.53,

729.53,

732.53,

735.53,

738.53,

741.53,

744.53,

747.54,

750.54,

753.55,

756.56,

759.56,

762.57,

765.58,

768.60,

771.61,

774.62,

777.64,

780.65,

783.67,

786.68,

789.70,

792.72,

795.74,

798.77,

801.79,

804.81,

807.84,

810.86,

813.89,

816.92,

819.95,

822.98,

826.01,

829.04,

832.07,

835.11,

838.14,

841.18,

844.22,

847.25,

850.29,

853.33,

856.37,

859.42,

862.46,

865.50,

868.55,

871.60,

874.64,

877.69,

880.74,

883.79,

886.84,

889.90,

892.95,

896.01,

899.06,

902.12,

905.18,

908.24,

911.30,

914.36,

917.42,

920.48,

923.55,

926.61,

929.68,

932.74,

935.81,

938.88,

941.95,

945.02,

948.10,

951.17,

954.24,

957.32,

960.40,

963.47,

966.55,

969.63,

972.71,

975.79,

978.88,

981.96,

985.05,

988.13,

991.22,

994.31,

997.40,

1000.49,

1003.58

}

wavelength(77)

%-- 05/12/2022 12:02 --%

AdjustCoordinates

%-- 06/12/2022 15:49 --%

AdjustCoordinates

coordinatesTable = [output\_point\_idn(:,1:5),output\_coordinates(:,1:3);output\_point\_idn(:,6:end),output\_coordinates(:,4:end);

];

coordinatesTable = [output\_point\_idn(:,1:5),output\_coordinates(:,1:3);output\_point\_idn(:,6:end),output\_coordinates(:,4:end)];

coordinatesTable = [output\_point\_idn(:,1:5),output\_coordinates(:,1:3)];

coordinatesTable = table('Size',[2\*n 8],'VariableTypes',{'double','string','string','double','double','double','double','double'},'VariableNames',["code","type","name","rank","group","X","Y","H"]);

coordinatesTable(1:n,:) = [output\_point\_idn(:,1:5),output\_coordinates(:,1:3)];

coordinatesTable(n+1:end,:) = [output\_point\_idn(:,6:end),output\_coordinates(:,4:end)];

TT = unique(coordinatesTable);

TT = unique(coordinatesTable,'rows');

TT.code(1)==TT.code(2)

TT.type(1)==TT.type(2)

TT.name(1)==TT.name(2)

TT.rank(1)==TT.rank(2)

TT(isnan(TT.code))=[];

rows=(isnan(TT.code));

TT(rows)=[];

TT(rows,:)=[];

rows = isnan(TT.rank);

TT.rank(rows) = 999;

A = sort(TT,1);

A = sortrows(TT,1);

A = sortrows(TT,4);

A = sortrows(TT,-1);

A = sortrows(TT,1,'descend');

A = sortrows(TT,1);

A = sortrows(TT,-1);

AdjustCoordinates

fileName

fileName(end-4:end)

fileName(1:end-4)

AdjustCoordinates

writetable(coordinatesTable,strcat('output\',fileName,'\_coordinates.csv'));

AdjustCoordinates

%-- 11/12/2022 10:48 --%

AdjustCoordinates

remainedInput = [mergedTableBad(:,1:14);mergedTableUnknown(:,1:14)];

remainedInput = [mergedTableBad(:,1:14);mergedTableUnknown(:,1:14)];

writetable(mergedTableGood,strcat('output\',fileName,'\_remained\_input.csv'));

writetable(remainedInput,strcat('output\',fileName,'\_remained\_input.csv'));

AdjustCoordinates

%-- 31/12/2022 16:34 --%

AdjustCoordinates

CSV2REZ

%-- 31/12/2022 16:58 --%

SwapDuplicateNames

inputFile = readtable(strcat(path,fileName),opts);

inputFile = readtable(strcat('output\_swapping\_duplicate\_names/',fileName),opts);

inputFile = readtable(strcat('output\_swapping\_duplicate\_names/',fileName));

inputFile = readtable(strcat('output\_swapping\_duplicate\_names/',fileName),'Delimeter',',');

inputFile = readtable(strcat('output\_swapping\_duplicate\_names/',fileName),'Delimiter',',');

opts = detectImportOptions(strcat(path,fileName),'Delimiter',',');

SwapDuplicateNames

rowsA = inputFile.typeA=='duplicate';

rowsA = string(inputFile.typeA)=='duplicate';

codesA = inputFile.codeA(rowsA);

mainNames = repmat(string(''),11,1);

A = '70F';

size(A,1)

A=NaN;

size(A,1)

A='NaN';

size(A,1)

SwapDuplicateNames

tempRows = string(allPointNames.type)=='min';

tempName = allPointNames.full\_name(tempRows);

size(tempName,1)

A=[1,0,2,0,0,4,0];

rows = A==0;

A(rows) = [5,5,5,5];

SwapDuplicateNames

inputFile.A(rowsA) = cell(namesA);

string2cell(A)

string2cell(namesA)

cell(namesA)

inputFile.A{rowsA} = namesA;

inputFile{rowsA,"A"} = namesA;

inputFile(rowsA,"A") = namesA;

inputFile.A(rowsA) = cellstr(namesA);

inputFile.typeA(rowsA) = 'main';

inputFile.typeA(rowsA) = "main";

inputFile.typeA(rowsA) = cellstr("main");

inputFile.groupA(rowsA) = groupsA;

SwapDuplicateNames

r = codesB==30284;

sum(r)

find(r)

r = codesA==30284;

sum(r)

SwapDuplicateNames

CSV2REZ

inputFile = inputFile(2:10);

inputFile = inputFile(:,2:10);

CSV2REZ

FindUnknownPoints

dh = inputFile.dH;

T=round(dh);

dh-T

mod(5.5,1)

mod(dH,1)

mod(dh,1)

sum(mod(dh,1))

FindUnknownPoints

%-- 01/01/2023 21:49 --%

SwapDuplicateNames

A = unique(inputFile(:,[2,3,4,5]),'rows');

%-- 03/01/2023 19:15 --%

FindUnknownPoints

wantedPoints = table(wanted\_points\_names');

wantedPoints = table(wanted\_points\_names');

wantedPoints.Properties.VariableNames = {'pointName'};

FindUnknownPoints

AdjustCoordinates

%-- 07/01/2023 14:24 --%

AdjustCoordinates

FindUnknownPoints

inputFile("B")

inputFile["B","dH"]

inputFile(:,["B","dH"])

inputFile(:,["A","dH"]) = inputFile(:,["B","diff"]);

inputFile.A{2} = 'A';

inputFile.A{3} = 'B';

inputFile.A{4} = 'C';

inputFile(:,["A","dH"]) = inputFile(:,["B","diff"]);

inputFile(:,["A","B"]) = inputFile(:,["B","A"]);

inputFile.A{2} = 'A';

inputFile.A{3} = 'B';

inputFile.A{4} = 'C';

inputFile(:,["A","B"]) = inputFile(:,["B","A"]);

inputFile(:,["A",-"dH"])

FindUnknownPoints

temp = SwapRows(connectedFromA)

FindUnknownPoints

temp = SwapRows(connectedFromA)

H = round(nanmean(tempH),3);

diff\_dH = round(nanmean(tempDiff\_dH),3);

FindUnknownPoints

DecreaseNumAcrossBy1

AdjustCoordinates

SwapDuplicateNames

CSV2WKT

CSV2REZ

A = GetCoordinates(readtable(strcat('output\_adjusting\_coordinates\','recent\_merged\_good.csv'));

A = GetCoordinates(readtable(strcat('output\_adjusting\_coordinates\','recent\_merged\_good.csv')));

SwapDuplicateNames

A = GetCoordinates(readtable(strcat('output\_adjusting\_coordinates\','recent\_merged\_good.csv')));

SwapDuplicateNames

input = readtable(strcat('output\_adjusting\_coordinates\','recent\_merged\_good.csv')));

input = readtable(strcat('output\_adjusting\_coordinates\','recent\_merged\_good.csv'));

opts = detectImportOptions('output\_adjusting\_coordinates\','recent\_merged\_good.csv'),'Delimiter',',');

opts = detectImportOptions(strcat('output\_adjusting\_coordinates\','recent\_merged\_good.csv'),'Delimiter',',');

input = readtable(strcat('output\_adjusting\_coordinates\','recent\_merged\_good.csv'),opts);

A=GetCoordinates(InputMergedTable);

A=GetCoordinates(input);

writetable(A,strcat('output\_swapping\_duplicate\_names/','temp\_coordinates.csv'));

CSV2REZ

SwapDuplicateNames

AdjustCoordinates

SwapDuplicateNames

CSV2WKT

CSV2REZ

AdjustCoordinates

%-- 16/01/2023 14:28 --%

AdjustCoordinates

UnifySegments

AdjustCoordinates

%-- 16/01/2023 16:06 --%

AdjustCoordinates

SwapDuplicateNames

CSV2WKT

CSV2REZ

UnifySegments

AdjustCoordinates

rows = (coordinates.group==8) && (coordinates.letter~='hsh') && (coordinates.letter~='pt') && (coordinates.letter~='h') && (coordinates.letter~='ra');

rows = (coordinates.group==8) & (coordinates.letter~='hsh') & (coordinates.letter~='pt') & (coordinates.letter~='h') & (coordinates.letter~='ra');

rows = (coordinates.group==8) & (string(coordinates.letter)~='hsh') & (string(coordinates.letter)~='pt') & (string(coordinates.letter)~='h') & (string(coordinates.letter)~='ra');

sum(rows)

sum(coordinates.group==8)

coordinates(rows,:)=[];

SwapDuplicateNames

CSV2WKT

CSV2REZ

AdjustCoordinates

SwapDuplicateNames

CSV2REZ

%-- 19/01/2023 11:31 --%

CSV2WKT

SwapDuplicateNames

CSV2WKT

%-- 21/01/2023 17:31 --%

AdjustCoordinates

SwapDuplicateNames

MergeDuplicates

A=[0 0 0 1 0 1];

find(A)

find(A,'first')

find(A,1,'first')

MergeDuplicates

t = inputFile(21225,:);

inputFile(counter,end-8:end)

inputFile(counter,end-7:end) + inputFile(indexes(i),end-7:end)

inputFile{counter,end-7:end} + inputFile{indexes(i),end-7:end}

inputFile(counter,end-7:end) = inputFile{counter,end-7:end} + inputFile{indexes(i),end-7:end};

inputFile{counter,end-7:end} = inputFile{counter,end-7:end} + inputFile{indexes(i),end-7:end};

writetable(inputFile,strcat(path,fileName,'\_merged\_duplicates.xlsx'));

MergeDuplicates

GetCoordinates

strsplit(fileName,'.')

GetCoordinates

GetCoordinatesFromMergedTable

temp{end}

GetCoordinatesFromMergedTable

CSV2WKT

%-- 23/01/2023 11:56 --%

UnifySegments

%-- 26/01/2023 20:21 --%

AdjustCoordinates

%-- 26/01/2023 20:34 --%

SwapDuplicateNames

CSV2WKT

CSV2REZ

%-- 26/03/2023 15:07 --%

X = 21:1.5:0

X = 21:-1.5:0

Y = 21:-1.5:0

[x,y] = meshgrid(X,Y)

x = 0:1.5:21;

y = 21:-1.5:0;

[X,Y] = meshgrid(x,y)

X=X-10.5;

Y=Y-10.5;

X = X + 131539.546;

Y = Y+568428.914;

N = 568443.106/Y(1,15)

acos(N)

asin(N)

atan(N)

[X,Y] = meshgrid(x,y)

X=X-10.5;

Y=Y-10.5;

X = X\*sin(0.487);

X = X + 131539.546;

[X,Y] = meshgrid(x,y)

X = X\*sin(0.487);

X=X-10.5;

X = X + 131539.546;

%-- 16/06/2023 17:13 --%

UnifySegments

%-- 17/06/2023 18:07 --%

AdjustCoordinates

%-- 18/06/2023 15:45 --%

untitled3

%-- 19/06/2023 11:03 --%

SwapDuplicateNames

CSV2WKT

CSV2REZ

%-- 01/07/2023 13:40 --%

[ E,N ] = IGD12toIG12( 31+56/60+30.98466/3600, 34+40/60+53.00806/3600, 0 )

[X,Y,Z]

[X\_GRS80,Y\_GRS80,Z\_GRS80]

[X\_WGS84,Y\_WGS84,Z\_WGS84] = GRS80toWGS84(4454994.96370864,3082564.53353631,3354935.21414808)

X = [4454994.96370864;4454994.96370864];

Y = [3082564.53353631;3082564.53353631];

Z=[3354935.21414808;3354935.21414808];

[X\_WGS84,Y\_WGS84,Z\_WGS84] = GRS80toWGS84(X,Y,Z)

T=([X\_GRS80;Y\_GRS80;Z\_GRS80]-[dX;dY;dZ]);

T=([X\_GRS80;Y\_GRS80;Z\_GRS80].-[dX;dY;dZ]);

[1 1]-1

[X\_WGS84,Y\_WGS84,Z\_WGS84] = GRS80toWGS84(X,Y,Z)

T=[X\_GRS80-dX;Y\_GRS80-dY;Z\_GRS80-dZ]

T=[X\_GRS80-dX,Y\_GRS80-dY,Z\_GRS80-dZ]';

M.\*T

T

[X\_GRS80-dX,Y\_GRS80-dY,Z\_GRS80-dZ]

ans'

M\*ans

M\*ans'

T=[X\_GRS80-dX,Y\_GRS80-dY,Z\_GRS80-dZ]

M\*T

M\*T'

ans'

[X\_WGS84,Y\_WGS84,Z\_WGS84] = GRS80toWGS84(X,Y,Z)

[ phi, lambda ] = IG12toIGD12(170000,650000 )

IG12toIGD12

[ phi, lambda ] = IG12toIGD12(170000,650000 )

IG12toIGD12

[ phi, lambda ] = IG12toIGD12(170000,650000 )

double(S.phi)

S.phi

[ phi, lambda ] = IG12toIGD12(170000,650000 )

(pi/2)\*180/pi

[ phi, lambda ] = IG12toIGD12(170000,650000 )

31+56/60+ 30.98466/3600

34+40/60+ 53.00806/3600

IGD12toIG12( 31.9419401833333, 34.6813911277778)

[E,N] = IGD12toIG12( 31.9419401833333, 34.6813911277778)

[E1,N1] = IGD12toIG12( 31.9419401772489, 34.6813911287901)

%-- 03/07/2023 10:19 --%

CreatePhiLambdaCoordinatesLists

%-- 08/07/2023 18:18 --%

CreateGraph

cycles{1}

string(cycles{1})

untitled

temp = [temp1;temp2];

invertedRows(:,["dH\_from\_DB","dH\_from\_file"])

T = inputFile(:,["dH\_from\_DB","dH\_from\_file"]);

S = -inputFile(:,["dH\_from\_DB","dH\_from\_file"]);

S = -inputFile{:,["dH\_from\_DB","dH\_from\_file"]};

ComputeLoops

sum(temp.dH)\*1000

CreateGraph

ComputeLoops

%-- 19/07/2023 08:07 --%

A = rand(2,2);

A(1,1)

%-- 22/07/2023 19:50 --%

ComputeLoops

%-- 31/07/2023 14:48 --%

ComputeLoops

%-- 06/08/2023 08:26 --%

ComputeLoops

[A,B] = cyclebasis(basisCycles{1})

T = basisCycles{3};

[A,B] = cyclebasis(T)

[A,B] = cyclebasis(basisCycles)

[A,B] = cyclebasis(basisCycles(1))

[A,B] = cyclebasis(basisCycles{1})

[A,B] = cyclebasis(basisCycles{2})

ComputeLoops

temp.Edges(1,"B")

ComputeLoops

temp.Edges{1,"B"}

temp.Edges(1,"B")

temp.Edges[1,"B"]

temp.Edges{1,"B"}

ComputeLoops

temp.Edges{1,"Direction"} = {1};

temp.Edges{1,"Direction"} = 1;

temp.Edges{1,"Direction"} = 5;

ComputeLoops

A=temp.Edges;

A=temp.Edges(5,:);

ComputeLoops

string(temp.Edges{1,"B"})

height(temp.Edges)

tempRow = temp.Edges(edgesOrder(j),:);

cell2mat(edgesOrder)

ComputeLoops

tempRow.A==lastPoint

tempRow.B==lastPoint

tempRow(1,"B")==lastPoint

tempRow.B(1)

tempRow.B{1}

temp.Edges(edgesOrder(j),:).B{1}

temp.Edges.B{edgesOrder(j)}

ComputeLoops

loopSummary.misc\_mm\_wo\_grav(i) = round(sum(tempEdges.dH.\*temp.Edges.Order)\*1000,2);

loopSummary.misc\_mm\_wo\_grav(i) = round(sum(temp.Edges.dH.\*temp.Edges.Order)\*1000,2);

clc

full(B)

sum(abs(ans(3,:)))

p = plot(temp,'EdgeLabel',temp.Edges.Weight);

p = plot(basisCycles{1},'EdgeLabel',basisCycles{1}.Edges.Weight);

p = plot(basisCycles{2},'EdgeLabel',basisCycles{2}.Edges.Weight);

p = plot(basisCycles{3},'EdgeLabel',basisCycles{3}.Edges.Weight);

p = plot(basisCycles{4},'EdgeLabel',basisCycles{4}.Edges.Weight);

p = plot(basisCycles{5},'EdgeLabel',basisCycles{5}.Edges.Weight);

p = plot(basisCycles{6},'EdgeLabel',basisCycles{6}.Edges.Weight);

p = plot(basisCycles{7},'EdgeLabel',basisCycles{7}.Edges.Weight);

p = plot(basisCycles{8},'EdgeLabel',basisCycles{8}.Edges.Weight);

p = plot(basisCycles{9},'EdgeLabel',basisCycles{8}.Edges.Weight);9

p = plot(basisCycles{9},'EdgeLabel',basisCycles{9}.Edges.Weight);

temp2 = subgraph(G,idx);

[A, edgesOrder] = cyclebasis(temp);

ComputeLoops

p = plot(basisCycles{1},'EdgeLabel',basisCycles{1}.Edges.Weight);

p = plot(basisCycles{2},'EdgeLabel',basisCycles{2}.Edges.Weight);

full(A)

ComputeLoops

p = plot(temp,'EdgeLabel',temp.Edges.Weight);

temp = rmedge(temp,8);

temp = rmedge(temp,3);

loopSummary = SummarizeLoops(temp);

loopSummary = SummarizeLoops({temp});

edgesOrder = cell2mat(edgesOrder);

lastPoint = string(temp.Edges{1,"B"})

temp.Edges{1,"Direction"} = 1;

height(temp.Edges)

temp.Edges(edgesOrder(2),:)

(temp.Edges.A{edgesOrder(j)}==lastPoint)

(temp.Edges.A{edgesOrder(2)}==lastPoint)

loopSummary = SummarizeLoops({temp});

tab=G.Edges(idx,:);

ComputeLoops

sum(B,'rows')

sum(B)

sum(B')

sum(abs(B'))

tab=G.Edges(idx,:);

M = graph(tab);

figure; p = plot(M);

p.XData = M.Edges.Y;

p.XData = unique(M.Edges.Y1;M.Edges.Y2);

p.XData = unique(M.Edges.Y1,M.Edges.Y2);

ComputeLoops

temp.Edges.A{edgesOrder(j)}==lastPoint

temp.Edges.A{edgesOrder(j)}

temp.Edges.A{edgesOrder(j)}==lastPoint

A=temp.Edges.A{edgesOrder(j)};

B=lastPoint;

A==B

size(A)

size(B)

string(A)==string(B)

string(A)==string(A)

(temp.Edges.A{edgesOrder(j)}==lastPoint)

clc

ComputeLoops

degree(G,'426A')

degree(G,'18U')

sum(abs(B'))

min(ans)

full(sum(abs(B')))

SD = B(157,:)

plot(temp)

Z=cyclebasis(temp)

ComputeLoops

M=setxor(temp)

help setxor

[cycles,edgecycles] = cyclebasis(temp);

M=setxor(edgecycles)

M=setxor(edgecycles{1},edgescycles{2})

M=setxor(edgecycles{1},edgecycles{2})

M=setxor(edgecycles{3},edgecycles{2})

M=setxor(edgecycles{3},edgecycles{2},edgecycles{3})

M=setxor(setxor(edgecycles{3},edgecycles{2}),edgecycles{3})

ComputeLoops\_bad

ComputeLoops

degree(G)

H=rmnode(G,degree(G)<2);

H=rmnode(G,find(degree(G)<2));

plot(H)

ComputeLoops

sum(x)

coordinatesY=[G.Edges.Y1;G.Edges.Y2];

coordinatesX=[G.Edges.X1;G.Edges.X2];

coordinatesAll=unique([coordinatesY,coordinatesX],'stable');

%-- 07/08/2023 10:13 --%

ComputeLoops

clc

ComputeLoops

S = rmnode(S,find(degree(S)==0));

clc

close all

plot(tempGLoop)

independentGLoops{removeIdx}={};

SD=independentGLoops{removeIdx};

SD=independentGLoops(removeIdx);

SD=independentGLoops[removeIdx];

SD=independentGLoops(1,removeIdx);

SD=independentGLoops(removeIdx);

SD=independentGLoops(logical(removeIdx));

SD=independentGLoops(logical(~removeIdx));

independentGLoops{logical(removeIdx)}={};

independentGLoops=independentGLoops(logical(~removeIdx));

independentGLoops = [independentGLoops;additionalGLoops];

independentGLoops = {independentGLoops;additionalGLoops};

U=[independentGLoops;additionalGLoops];

clc

plot(independentGLoops2)

plot(independentGLoops2{7})

graph(G.Edges(edgecycles{i}),G.Nodes.Names==cycles{i});

graph(G.Edges(edgecycles{i}));

L=G.Edges(edgecycles{i});

L=G.Edges(edgecycles{i,:});

L=G.Edges;

L(edgecycles(1))

L(edgecycles{1})

edgecycles{1}

M=edgecycles{1}

L(M)

L{M}

L{M,:}

L(1,1)

L(1,:)

L([1,2],:)

L(M',:)

L(M,:)

L(edgecycles{i},:)

G.Edges(edgecycles{i},:)

G.Nodes.Names==cycles{i}

G.Nodes.Name==cycles{i}

G.Nodes.Name==string(cycles{i})

G.Nodes.Name==string(cycles{i}')

[~,ind] = ismember(cycles{i}, G.Nodes.Name);

ind

independentGLoops2{i} = graph(G.Edges(edgecycles{i},:),G.Nodes(ind,:));

ComputeLoops

temp = independentGLoops(7);

temp = independentGLoops2(7);

temp = independentGLoops2{7};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

sum(temp.Edges.dist)

temp = independentGLoops2(6);

hold on

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

temp = independentGLoops2{6};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

hold on

temp = independentGLoops2{7};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

temp = independentGLoops2{19};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

hold on

temp = independentGLoops2{6};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

hold on

temp = independentGLoops2{16};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

hold on

temp = independentGLoops2{128};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

hold on

temp = independentGLoops2{19};

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

loopSummary2 = sortrows(loopSummary2,"length\_km","descend");

hold on

temp = independentGLoops2{27};

loopSummary2 = sortrows(loopSummary2,"length\_km","descend");

loopSummary2 = sortrows(loopSummary2,"number","ascend");

temp = independentGLoops2{27};

hold on

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

for i=1:323

temp = independentGLoops2{i};

hold on

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

end

figure

for i=1:302

temp = independentGLoops2{i};

hold on

plot(temp,'EdgeLabel',temp.Edges.Weight,'XData',temp.Nodes.Y,'YData',temp.Nodes.X);

end

%-- 09/08/2023 12:58 --%

ComputeLoops

gTable = table(inputGrav.g(ind),'VariableNames',{'g'});

inputFileCoordinates = [inputFileCoordinates,gTable];

inputFileCoordinates = removevars(inputFileCoordinates, "g");

inputFileCoordinates = [inputFileCoordinates,gTable];

sum(isnan(gTable.g))

ComputeLoops

inputFileCoordinates(inputFileCoordinates.rank==0,:)=[];

inputFileCoordinates = GetCoordinatesFromMergedTable(inputFile);

inputFileCoordinates = [inputFileCoordinates,gTable];

inputFileCoordinates.rank(inputFileCoordinates.rank==0)=[];

inputFileCoordinates.rank(inputFileCoordinates.rank==0)=NaN;

inputFileCoordinates.H(inputFileCoordinates.H==0)=NaN;

A=inputFileCoordinates.rank(inputFileCoordinates.rank==2);

ComputeLoops

%-- 05/11/2023 10:44 --%

ComputeLoops

%-- 26/11/2023 14:09 --%

AdjustCoordinates

SwapDuplicateNames

CSV2WKT

CSV2REZ

ComputeLoops

CreateGraph

Copy\_of\_ComputeLoops

%-- 26/11/2023 15:45 --%

ComputeLoops

%-- 27/11/2023 17:56 --%

AdjustCoordinates

coordinates(coordinates.group==8,:)=[];

T = coordinates(coordinates.group==8 && coordinates.letter=='hsh',:);

T = coordinates(coordinates.group==8 & coordinates.letter=='hsh',:);

T = coordinates(coordinates.group=8 & coordinates.letter='hsh',:);

coordinates(coordinates.group==11,:)=[];

T = coordinates(coordinates.group==8,:);

T = coordinates(coordinates.group==9,:);

T = coordinates((coordinates.group==9 & coordinates.letter=='A') ,:);