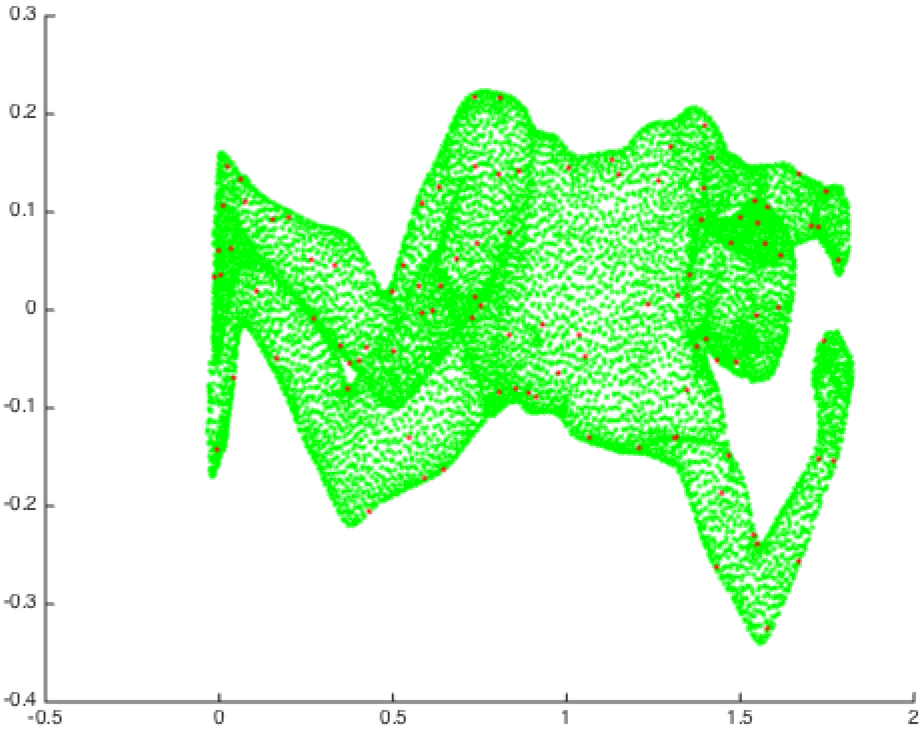
Table of Contents

```
clear;
close all;
load('mesh_hks.mat');
%%============read mesh from the off file============%%
```

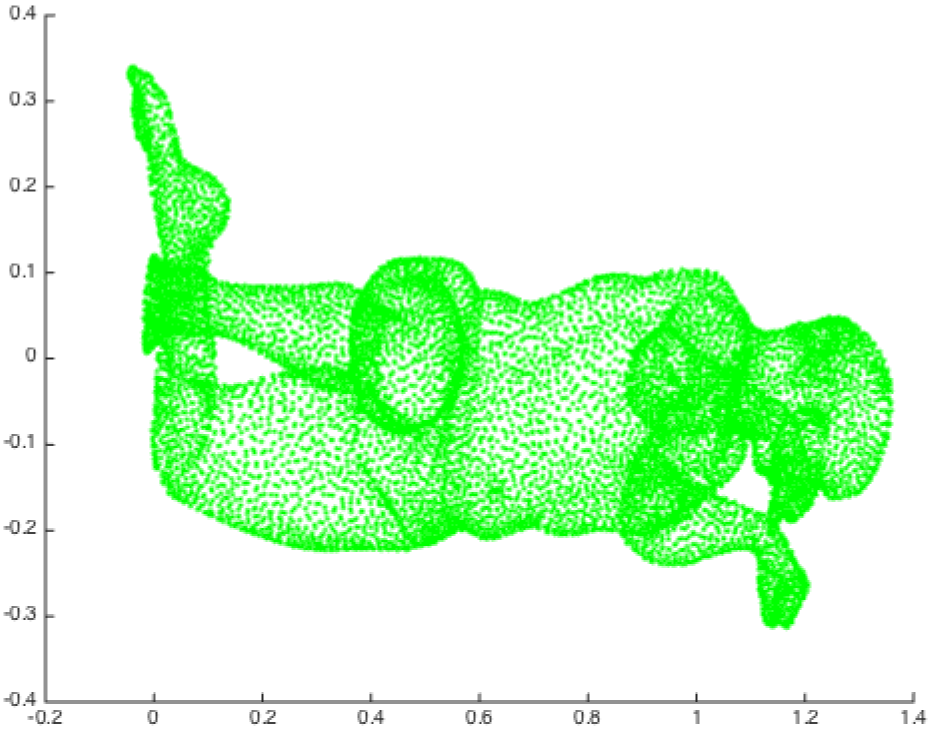
DATA1

```
fid=fopen('/Users/student/Downloads/computer_vision_HW3/mesh015.off');
fgetl(fid);
nos = fscanf(fid, '%d %d  %d', [3 1]);
nopts = nos(1);
notrg = nos(2);
coord = fscanf(fid, '%g %g  %g', [3 nopts]);
coord = coord';
triang=fscanf(fid, '%d %d %d %d',[4 notrg]);
triang=triang';
triang=triang(:,2:4)+1;
%%we have added 1 because the vertex indices start from 0 in vtk format
fclose(fid);
hold on;
plot3(coord(:,1),coord(:,2),coord(:,3),'g.');
plot3(coord(1:125:12500,1),coord(1:125:12500,2),coord(1:125:12500,3),'r.');
mesh015_ks = mesh015_hks(1:125:12500,:);
hold off
descriptorgeodesic = zeros(12500,12500);
    [v,c] = size(triang);
    for i = 1:v
        descriptorgeodesic(triang(i,1),triang(i,2)) = sqrt(sum((coord(triang(i,2),
        descriptorgeodesic(triang(i,1),triang(i,3)) = sqrt(sum((coord(triang(i,3),
        descriptorgeodesic(triang(i,2),triang(i,3)) = sqrt(sum((coord(triang(i,3),
        descriptorgeodesic(triang(i,2),triang(i,1)) = descriptorgeodesic(triang(i,
        descriptorgeodesic(triang(i,3),triang(i,1)) = descriptorgeodesic(triang(i,
        descriptorgeodesic(triang(i,3),triang(i,2)) = descriptorgeodesic(triang(i,
    end
```



DATA2

```
fid=fopen('/Users/student/Downloads/computer_vision_HW3/mesh054.off');
fgetl(fid);
nos1 = fscanf(fid, '%d %d  %d', [3 1]);
nopts1 = nos(1);
notrg1 = nos(2);
coord1 = fscanf(fid, '%g %g  %g', [3 nopts]);
coord1 = coord1';
triang1=fscanf(fid, '%d %d %d %d',[4 notrg]);
triang1=triang1';
triang1=triang1(:,2:4)+1;
%we have added 1 because the vertex indices start from 0 in vtk format
fclose(fid);
figure(2)
hold on
plot3(coord1(:,1),coord1(:,2),coord1(:,3),'g.');
plot3(coord1(1:125:12500,1),coord1(1:125:12500,2),coord1(1:125:12500,3),'r.');
mesh054_ks = mesh054_hks(1:125:12500,:);
hold off;
 descriptorgeodesic1 = zeros(12500,12500);
    [v,c] = size(triang1);
    for i = 1:v
        descriptorgeodesic1(triang1(i,1),triang1(i,2)) = sqrt(sum((coord1(triang1(
        descriptorgeodesic1(triang1(i,1),triang1(i,3)) = sqrt(sum((coord1(triang1(
            descriptorgeodesic1(triang1(i,2),triang1(i,3)) = sqrt(sum((coord1(triang1(
            descriptorgeodesic1(triang1(i,2),triang1(i,1)) = descriptorgeodesic1(trian
            descriptorgeodesic1(triang1(i,3),triang1(i,1)) = descriptorgeodesic1(trian
            descriptorgeodesic1(triang1(i,3),triang1(i,2)) = descriptorgeodesic1(trian
        end
```



data action - assignment - using hungarian algorithm

```
[n1,m1] = size(mesh015_hks);
[n2,m2] = size(mesh054_hks);
MATRIX1 = zeros(100,100);
for i = 1:100
    MATRIX1(i,:) = sqrt(sum((repmat(mesh015_ks(i,:),100,1) - mesh054_ks).^2,2))';
end
m = min(MATRIX1,[],2);

%STEP1
MATRIX = MATRIX1 - repmat(m,1,100);
MATRIX2 = MATRIX;
%[m1,I] = min(MATRIX,[],1);
%MATRIX = MATRIX - repmat(m1,100,1);
tickr = zeros(1,100);
tickc = zeros(1,100);
assignedr = zeros(1,100);
```

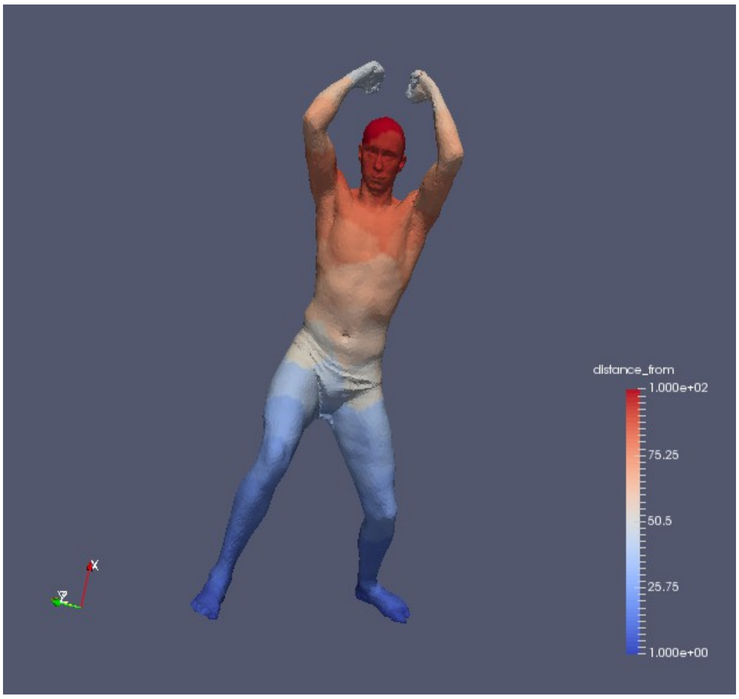
```
maskc = zeros(1,100);
[R,C] = find(MATRIX == 0);
MASK = zeros(100,100);
for i = 1:100
    k = find(MATRIX(i,:) == 0);
    k1 = find(maskc == 1);
    k = setdiff(k,k1);
    if(assignedr(i) == 0 && not(isempty(k)))
        MASK(i,k(1)) = 1;
        assignedr(i) = 1;
        maskc(k(1)) = 1;
    end


end
assignedr = zeros(1,100);
maskc = zeros(1,100);
done = false;
while(not(done))
%step3
[I,K] = find(MASK == 1);
maskc(K) = 1;
if (sum(maskc) == 100)
   done = true;
   break;
end






done1 = false;
unmaskedc = find(maskc == 0);
while(not(done1))
%step 4
unmaskedc = find(maskc == 0);
unmaskedr = find(assignedr == 0);
[I,K] = find(MATRIX(unmaskedr,unmaskedc) == 0);

 if(not(isempty(K)))
    i = unmaskedr(I(1));
    j = unmaskedc(K(1));
    MASK(i,j) = 2;
    K12   = find(MASK(i,:) == 1);
    if (not(isempty(K12)))
        %[I1,K1]   = find(MASK(I,:) == 1);
        assignedr(i) = 1;
        maskc(K12) = 0;
        step = 6;
    else

        step = 5;
        uncoveredr = i;
        uncoveredc = j;
        done1 = true;

    end
 else
     %step = 6;
     MINVAL = min(min(MATRIX(unmaskedr,unmaskedc)));
     rmask = find(assignedr == 1);
     MATRIX(rmask,:) = MATRIX(rmask,:)+MINVAL;
     MATRIX(:,unmaskedc) = MATRIX(:,unmaskedc) - MINVAL;
 end

end
%step5
k = find(MASK(:,uncoveredc) == 1);
MASK(uncoveredr,uncoveredc) = 1;
L1 = uncoveredc;
i = 1;
while(not(isempty(k)))
L = find(MASK(k,:) ==2);
MASK(k,L1) = 0;
k1 = k;
k = find(MASK(:,L) == 1);
MASK(k1,L) = 1;
L1 = L;
end
end

assignmentmatrix = zeros(100,100);
assignmentmatrix(find(MASK == 1)) = 1;
COST = sum(MATRIX1(find(assignmentmatrix==1)));
```

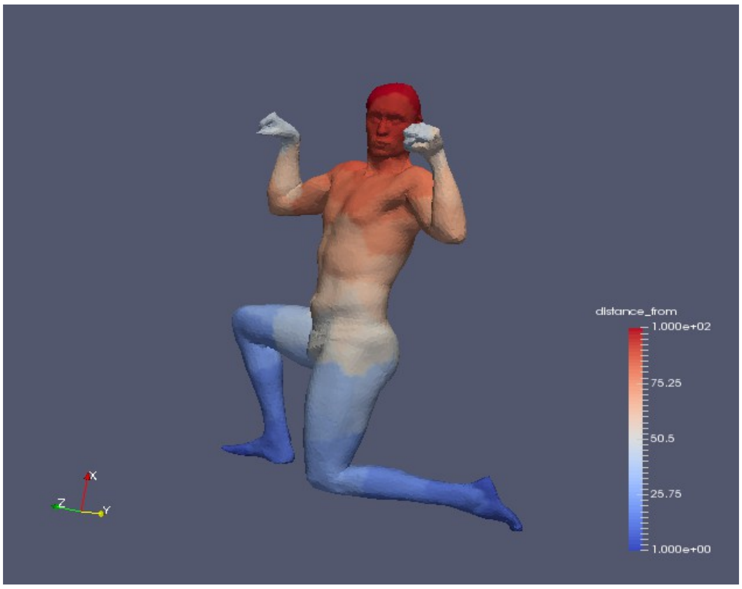
COLORMAP CREATION

```
LIST = [1:125:12500];
correspond = zeros(1,100);
source = 1;
L1 = zeros(100,12500);
L2 = zeros(100,12500);
for source = 1:1:100
L1(source,:) = graphshortestpath(sparse(descriptorgeodesic),LIST(source));
source1 = find(assignmentmatrix(source,:)== 1);
correspond(source1) = source;
L2(source1,:) = graphshortestpath(sparse(descriptorgeodesic1),LIST(source));
end
closeness = [1:1:100];
[Min,I] = min(L1,[],1);
f_vec1 = closeness(I);
[Min,I] = min(L2,[],1);
f_vec2 = closeness(correspond(I));

ofid = fopen('man1.vtk','w');
    fprintf(ofid, '# vtk DataFile Version 3.0\n');
fprintf(ofid,'vtk output\n');
fprintf(ofid,'ASCII\n');
fprintf(ofid,'DATASET POLYDATA\n');
fprintf(ofid,'POINTS %d float\n', nopts);
fprintf(ofid,'%g %g %g\n', coord');
fprintf(ofid,'POLYGONS %d %d\n', notrg, 4*notrg);
fprintf(ofid,'3 %d %d %d\n', triang'-1);
fprintf(ofid,'\n');
fprintf(ofid,'POINT_DATA %d\n', nopts);
fprintf(ofid,'SCALARS distance_from float\n');
fprintf(ofid,'LOOKUP_TABLE default\n');
fprintf(ofid,'%g\n',  f_vec1');
fclose(ofid);

ofid = fopen('man2.vtk','w');
fprintf(ofid, '# vtk DataFile Version 3.0\n');
fprintf(ofid,'vtk output\n');
fprintf(ofid,'ASCII\n');
fprintf(ofid,'DATASET POLYDATA\n');
fprintf(ofid,'POINTS %d float\n', nopts1);
fprintf(ofid,'%g %g %g\n', coord1');
fprintf(ofid,'POLYGONS %d %d\n', notrg1, 4*notrg1);
fprintf(ofid,'3 %d %d %d\n', triang1'-1);
fprintf(ofid,'\n');
fprintf(ofid,'POINT_DATA %d\n', nopts1);
fprintf(ofid,'SCALARS distance_from float\n');
fprintf(ofid,'LOOKUP_TABLE default\n');
fprintf(ofid,'%g\n',  f_vec2');
fclose(ofid);
```

Published with MATLAB® R2014b

OUTPUT


distance_from
1.000e+02
75.25
50.5
25.75
1.000e+00
X
Z


distance_from
1.000e+02
75.25
50.5
25.75
1.000e+00
X
Z
Y