3D reconstruction code:

https://github.com/davidt0x/micro-recon

Code Execution Guideline:

Open Visual Studio:

Open solution (solid_opt) in visual studio:

```
Right click on each of them one by one OptimizeSolidMEX,
GetAllNHoodsMEX, Extract3DNeighborhoods and do the following changes:

DEBUG -> OptimizeSolidMEX properties -> VC++ Directories:
Add $(MATLAB_ROOT)\extern\include; to Include Directories

DEBUG -> OptimizeSolidMEX properties -> Linker -> General:
Add $(MATLAB_ROOT)\extern\lib\win64\microsoft;
to Additional Library Directories
```

Manually change
MATLAB\R2012a\extern\include
to
MATLAB\2014b\extern\include

BUILD-> Build Solution

Open Matlab:

//add_paths: adds the path of flann_search directory in matlab
add_paths
//check if flann_seach is in the path
flann_search
//check if OptimzeSolidMEX is in the path
OptimizeSolidMEX
//Generate a random cube of dim 100x100x100
M = round(rand(100,100,100));
//Read zebra image to use as an exemplar
I = imread('../../images/zebra.png');
imshow(I);
//this program only accepts black and white input, so convert zebra image to black and white from //RGB. Select three exemplar. For the sake of simplicity, select identical exemplar.
I = imread('../../images/zebra.png'); I = double(I); I = I ./ 255; E{1} = I; E{2} = I; E{3} = I;

imshow(I);
edit SetupReconMultiRes.m

//input parameters for reconstruction. Exempler array, number of levels of reconstruction, //neighborhood size for each level of reconstruction, iteration range, number of iterations at each //level of reconstruction, ??

ReconH = SetupReconMultiRes(E, 3, , 1:3, [100 100 100], 4);

```
//make necessary changes??
edit SetupRecon

ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);

//re-run reconstruction, if you get a segmentation fault, but first release the memory
used.
DestroyReconMultiRes(ReconH);

//re-run reconstruction.
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);

//make necessary changes??
edit SolidOptimization.m

//run the script
RunIt
%-- 7/21/2015 3:24 PM --%
I = imread('../../images/zebra.png'); I = double(I); I = I ./ 255; E{1} = I; E{2} = I; E{3} = I;
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
DestroyReconMultiRes(ReconH);
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
RunIt
edit PlotIteration.m
RunIt
%-- 7/21/2015 3:32 PM --%
//enable mex file debugging. Downside: makes the execution of program slower
dbmex on
I = imread('../../images/zebra.png'); I = double(I); I = I ./ 255; E{1} = I; E{2} = I; E{3} = I;
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
RunIt
%-- 7/21/2015 3:40 PM --%
edit TestRun
TestRun
%-- 7/21/2015 3:45 PM --%
TestRun

//clears mex function, if it is not locked.
clear mex
//run the script that reads an image, converts from rgb to black&white and finally
reconstructs the
//image
TestRun
%-- 7/21/2015 3:57 PM --%
TestRun

//Debugging seg fault
edit resize
edit RunIt
RunIt
```

```
ii
ReconH{3}
ReconH{3}.TexelWeights
resize(ReconH{ii}.TexelWeights{pp}, size(ReconH{ii-1}.E{pp}), 'nearest');
size(ReconH{ii-1}.E{pp}
size(ReconH{ii-1}.E{pp})
E{1}
size(ReconH{ii-1}.E{pp})
E
return
DestroyReconMultiRes(ReconH);
size(I)
I = imread('../../images/zebra.png');
TestRun
DestroyReconMultiRes(ReconH);
I = imread('../../images/zebra.png'); I = double(I); I = I ./ 255; E{1} = I; E{2} = I;
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:2, [100 100 100], 4);
size(E)
clear E
I = imread('../../images/zebra.png'); I = double(I); I = I ./ 255; E{1} = I; E{2} = I;
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:2, [100 100 100], 4);
RunIt
I = imread('../../images/zebra.png'); I = double(I); I = I(:.:.1) ./ 255; E{1} = I; E{2} = I;
I = imread('../../images/zebra.png'); I = double(I); I = I(:,:,1) ./ 255; E{1} = I; E{2} = I;
clear E
I = imread('../../images/zebra.png'); I = double(I); I = I(:,:,1) ./ 255; E{1} = I; E{2} = I;
DestroyReconMultiRes(ReconH);
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:2, [100 100 100], 4);
RunIt
DestroyReconMultiRes(ReconH);
I = imread('../../images/corral.png'); I = double(I); I = I(:,:,1) ./ 255; E{1} = I; E{2} = I;
imshow(I);
close all
imshow(I)
size(I)
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
I = imread('../../images/corral.png'); I = double(I); I = I(:,:,1) ./ 255; E{1} = I; E{2} = I; E{3}
= I;
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
RunIt
max(I(:))
min(I(:))
mean(I(:))
imshow(I)\
imshow(I(:))
close all
imshow(I(:))
imshow(I)
RunIt
DestroyReconMultiRes(ReconH)
```

```
I = imread('../../images/corral.png'); I = double(I); I = I(:,:,1) ./ 255; E{1} = I; E{2} = I; E{3}
= I;
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
imshow(I)
RunIt
DestroyReconMultiRes(ReconH)
clear(E);
clear E
I = imread('../../images/zebra.png'); I = double(I); I = I(:,:,1) ./ 255; E{1} = I; E{2} = I; E{3}
= I;
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
I = imread('../../images/zebra.png'); I = double(I); I = I(:,:,1) ./ 255; E{1} = I; E{2} = I; E{3}
= I;
imshow(I)
RunIt
```

Errors encountered:

```
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
Warning: Name is nonexistent or not a directory:
C:\Users\ahafeez6\Documents\GitHub\micro-recon\cpp\bin
> In path at 109
 In addpath at 86
 In add_paths at 2
 In SetupReconMultiRes at 4
Level 1
        Getting neighborhoods for exemplar 1 of 3 ...Undefined function
'GetAllNHoodsMEX' for input arguments of type 'double'.
```

Solution:
The directory from which files were copied isn't in the path

```
add ..\GitHub\micro-recon\src\matlab
```

Error:

```
ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
Level 1
        Getting neighborhoods for exemplar 1 of 3 ...Undefined function
'GetAllNHoodsMEX' for input arguments of type 'double'.
```

Solution:

```
add ..\GitHub\micro-recon\src\cpp\bin
```

You need to make sure GetAllNHoodsMEX.mexw64 is in the directory you added to your path. If it is, then when you type GetAllNHoodsMEX and hit enter it should not say it is missing function.

Error:

```
>> ReconH = SetupReconMultiRes(E, 3, [9 9 9], 1:3, [100 100 100], 4);
Level 1
```

Getting neighborhoods for exemplar 1 of 3 ...Done
Building FLANN index for exemplar 1 of 3 ...Undefined function 'flann_build_index' for input arguments of type 'struct'.

Solution:

add ..\GitHub\micro-recon\src\flann

Error:
Building FLANN index for exemplar 3 of 3 ...Done. Time to Build = 0.253474 seconds
Undefined function 'twopointnp2' for input arguments of type 'double'.

Error in SetupRecon (line 168)
[G D] = twopointnp2(RS.E{1});

Error in SetupReconMultiRes (line 41)
ReconHierarchy{level} = SetupRecon(Exyz, NB_SIZES(level), NB_INDICIES, RECON_SIZE, NUM_CORES);

Solution:
download it from email and add twopointnp2 code in the path

Error:
>> RunIt
Iteration: 1
Finding nearest neighbors ... Done: 0.540831
Optimizing solid ... Undefined function or variable 'freezeColors'.

Error in PlotIteration (line 14)
freezeColors;

Solution:
download freezColors from matlab exchange and add it into the path.