Supplementary Material

1. MATLAB snippet for removal for Edge Enhancement Effect in gas hydrate datasets

1.1. Gas Hydrate Segmentation

1.1. Step 1

The Dual Clustering approach, by which first the 16-bit gas hydrate was filtered using Anisotropic diffusion (AD) filter, and then with non-local means (NLM) filter, to minimize/normalize the edge enhancement (ED) artefacts.

1.2. Step 2

- read slice by slice 3D prefiltered raw data
- for this example the reading is restricted
- to only four slices (700x700x4); it can be changed using nZ variable

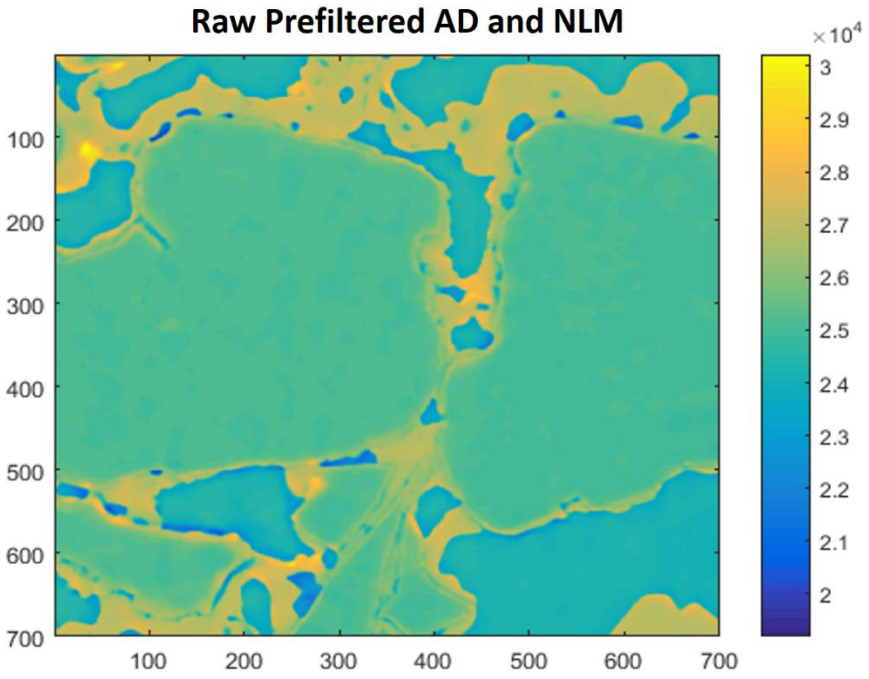
```
mfname= 'GasHydrat_ROI4_AD_NLM';
ifname=[mfname,'.raw'];
nX=700;
nY=700;
nZ=4;
ldim = 1;
xDi=[nX nY]';
grenzwert=0;
clusterS =7;
ifid=fopen(ifname,'r');
M=zeros(nX,nY,nZ,'uint16');
SeData = nZ-ldim;
SeData  = 1:1:SeData;
dim = size(M);


for k=ldim:nZ
    disp(sprintf('Reading slice no. %d....',k));
    s=sprintf('Slice: % d', k');
    S=fread(ifid, [xDi(1) xDi(2)], 'uint16');
    M(:,:,k)=S;
    %figure; imagesc(M(:,:,k)); colorbar;
end
```

```
Reading slice no. 1....
Reading slice no. 2....
Reading slice no. 3....
Reading slice no. 4....
```

1.3. Display image

```
figure; imagesc(M(:,:,1)); colorbar;
title('Raw Prefiltered ASD NLM')
```



1.4. Concatenate raw data into single array

```
%* concatenate array will be used in step three

M = M(:,:,ldim:nZ);
rawM = double(M(:));
```

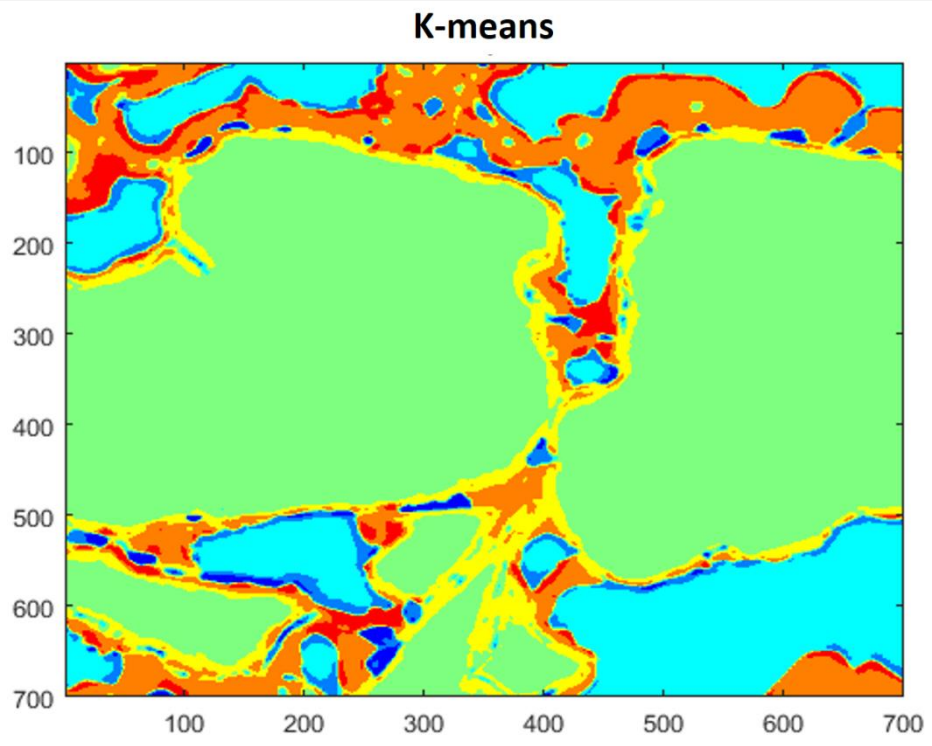
1.5. Perform k-means clustering

Here, clustering is restricted to class 7 optimal to enable clustering of all the available features:

```
for ii = 1:2
    R=double(M(:,:,ii));
    [r,c,v]=find(R>grenzwert);
    cyl=R>grenzwert;
    R1=cyl.*R;
    [m, n, w]=find(R1);
    G = kmeansK(w,clusterS);
    S=sparse(r,c,G,size(R,1),size(R,2));
    M_seg=full(S);
    SegImg(:,:,ii)=M_seg;
    %figure; imagesc(SegImg(:,:,ii)); colormap(parula(5)); colorbar;
    %title('K-means prefiltered');
end
```

1.6. Display image

```
figure; h = imagesc(SegImg(:,:,1)); colormap(jet(max(h.CData(:))));
title('K-means');
```



1.7. Step three

The purpose is to |index out| pixel values of different phases:

```
% noise
% edge enhanced low (EDL)
% liquid
% quartz
% edge enhanced high (EDL)
% gas hydrate
% from the concatenated raw images matrix using segmented class values
% thereafter compare their histogram % _as sanity check_
% to identify if any overlapping boundaries
```

1.8. Index noise pixels

```
rangeNl = 0;
indN = find(h.CData(:)==rangeNl);
rawO = rawM(indN);
```

1.9. Plot histogram noise

```
[cN, countN] = hist(rawO, 10);
%figure; bar(countN, cN);
%title('noise')
```

1.10. Index EDL pixels

```
rangeNu = 2;
indD = find(h.CData(:)>rangeNl & h.CData(:)<=rangeNu);
rawD = rawM(indD);
```

1.11. Plot histogram noise

```
[cD, countD] = hist(rawD, 100);
%figure; bar(countD, cD);
%title('Edge Enhanced low noise')
```

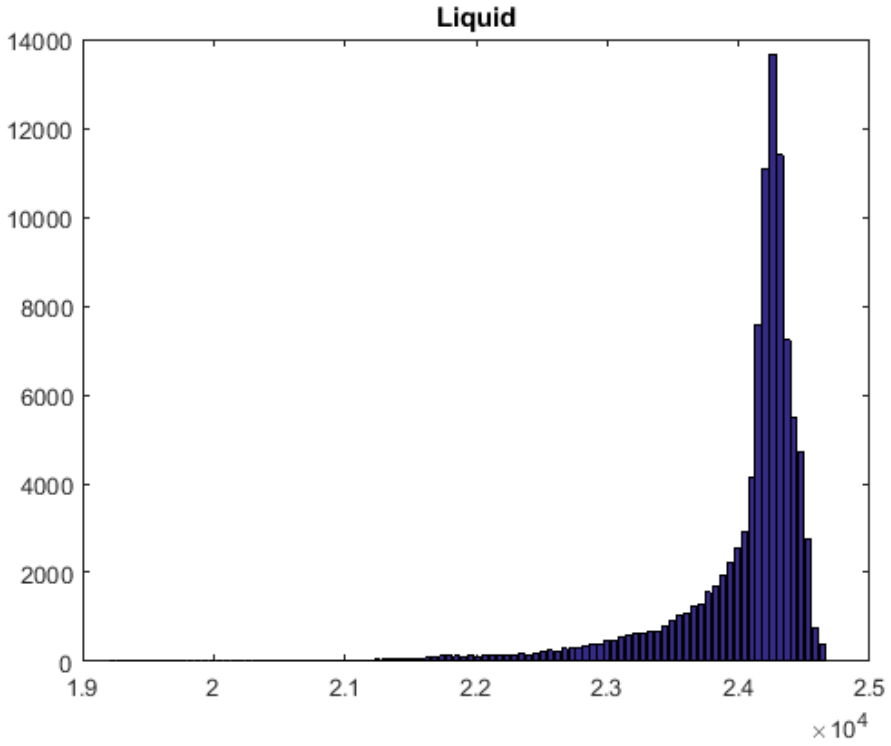
1.12. Index liquid pixels

```
rangeLl = 1;
rangeLu = 3;

indL = find(h.CData(:)>=rangeLl & h.CData(:)<=rangeLu);
if min(SegImg(indL))==rangeLl & max(SegImg(indL))==rangeLu
    rawL = rawM(indL);
    min_rawL = min(rawL);
    max_rawL = max(rawL);
    Avg_rawL = mean(rawL);
else
    fprintf('min and max for liquid dont match.....\n')
    return
end
```

1.13. Plot histogram liquid

```
[cL, countL] = hist(rawL, 100);
figure; bar(countL, cL);
title('Liquid')
```



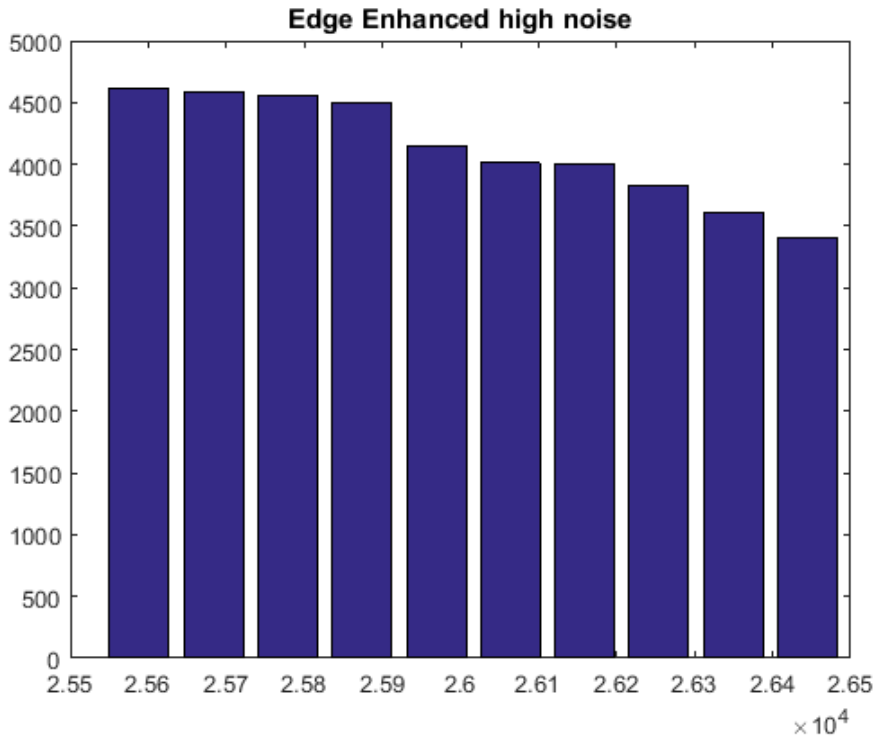
1.14. Index EDH pixels

```
rangeE = 5;
indE = find(h.CData(:)==rangeE);
if min(SegImg(indE))==rangeE
    rawE = rawM(indE);
    min_rawE = min(rawE);
    max_rawE = max(rawE);
    Avg_rawE = mean(rawE);
else
    fprintf('min and max for EDH dont match.....\n')
```

```
    return
end
```

1.15. Plot histogram EDH

```
[cE, countE] = hist(rawE, 10);
figure; bar(countE, cE);
title('Edge Enhanced high noise')
```



1.16. Quartz index phases

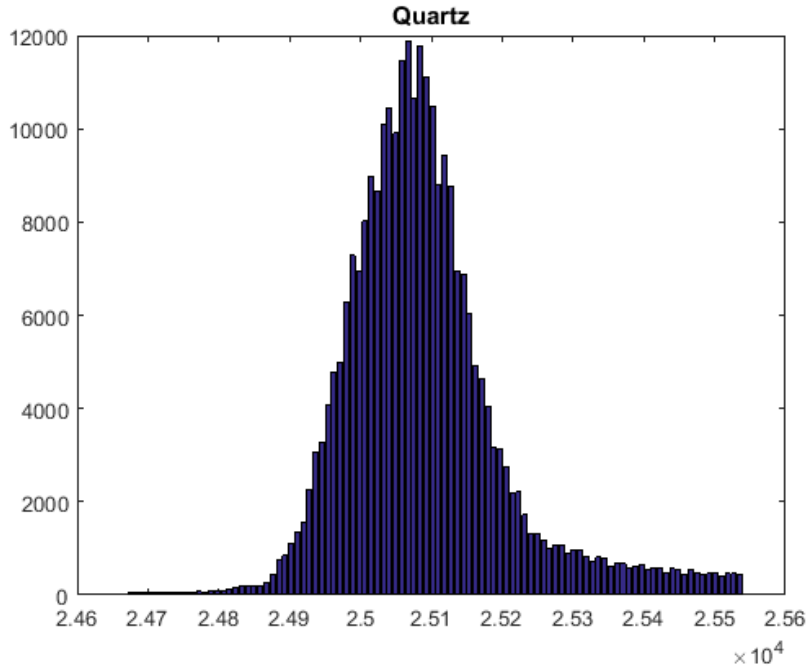
```
rangeQu = 4;
```

1.17. Quartz

```
indQ = find(h.CData(:)==rangeQu);
if min(SegImg(indQ)) == rangeQu
    rawQ = rawM(indQ);
    min_rawQ = min(rawQ);
    max_rawQ = max(rawQ);
    Avg_rawQ = mean(rawQ);
else
    fprintf('min and max for quartz dont match.....\n')
    return
end
%indQ = find(h.CData(:)>=rangeQl & h.CData(:)<=rangeQu);
```

1.18. Plot histogram quartz

```
[cQ, countQ] = hist(rawQ, 100);
figure; bar(countQ, cQ);
title('Quartz')
```

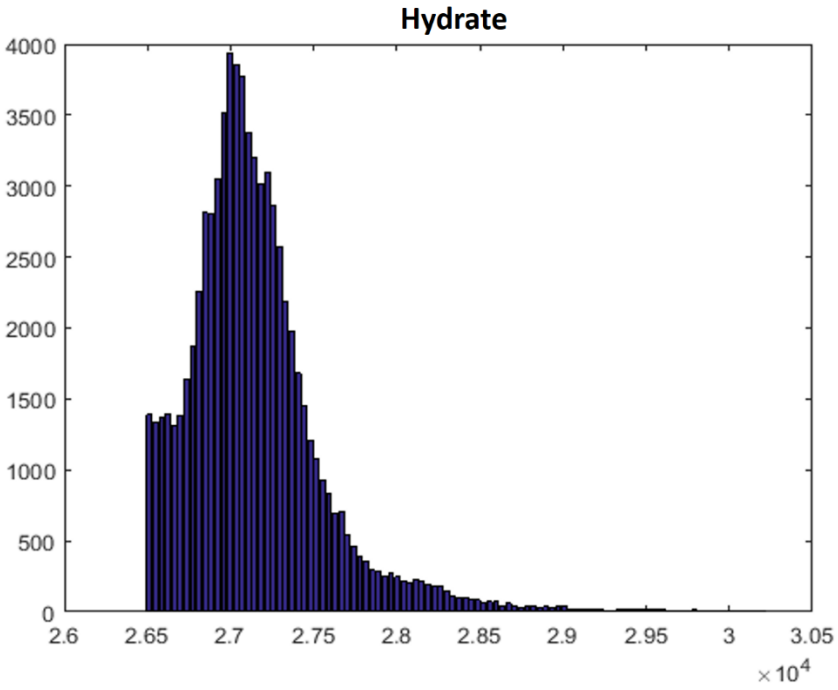


1.19. Gas Hydrate

```
rangeMl =6;
rangeMu =7;
%indM =find(h.CData(:)>=rangeMu);
indM = find(h.CData(:)>=rangeMl & h.CData(:)<=rangeMu);
if min(SegImg(indM))==rangeMl & max(SegImg(indM))==rangeMu
    rawMu = rawM(indM);
    min_rawMu = min(rawMu);
    max_rawMu = max(rawMu);
    Avg_rawMu = mean(rawMu);
elseif min(SegImg(indM))==rangeMu & max(SegImg(indM))==rangeMu
    rawMu = rawM(indM);
    min_rawMu = min(rawMu);
    max_rawMu = max(rawMu);
    Avg_rawMu = mean(rawMu);
else
    fprintf('min and max for gas hydrate dont match.....\n')
    return
end
```

1.20. Plot Histogram Gas Hydrate

```
[cM, countM] = hist(rawMu, 100);
figure; bar(countM, cM);
title('Hydrate')
```



1.21. Step 4 - rescaling the raw images

First min-max and mean of respective phases are calculated for the respective (raw) phases (obtained above), which thereafter are replaced by their respective mean values.

```
%average values
```

1.22. With an exception to GH dataset

Where EDH (raw pixels) are replaced with averaged quartz values:

```
%as they are in close vicinity to quartz pixel values

M_replace = M(:);
min_li = min_rawL;
max_li =max_rawL;
avg_li =Avg_rawL;
min_Qz = min_rawQ;
max_Qz = max_rawQ;
avg_Qz = Avg_rawQ;
min_EDH = min_rawE;
max_EDH = max_rawE;
avg_EDH = Avg_rawE;
min_GH = min_rawMu;
max_GH = max_rawMu;
avg_GH = Avg_rawMu;

%-------------------------------------------------------------------------
% indxes of liquid pixels
%-------------------------------------------------------------------------
Ind_rep_L = find(M_replace>=min_li & M_replace <= max_li);
% replacement by average liquid value
if min(M_replace(Ind_rep_L))==min_li & max(M_replace(Ind_rep_L)==max_li)
    M_replace(Ind_rep_L)=avg_li;
else
    fprintf('min and max for liquid dont match.....\n')
    return
end

%-------------------------------------------------------------------------
% indxes of quartz pixels
%-------------------------------------------------------------------------
Ind_rep_Q = find(M_replace>= min_Qz & M_replace<= max_Qz);
% replacement by average quartz value
if min(M_replace(Ind_rep_Q))==min_Qz & max(M_replace(Ind_rep_Q))==max_Qz
    M_replace(Ind_rep_Q)= avg_Qz;
else
    fprintf('min and max for quartz dont match.....\n')
    return
end
```

```
%--------------------------------------------------------------------------
% indxes of EDH pxels
%--------------------------------------------------------------------------
Ind_rep_E = find(M_replace>= min_EDH & M_replace<=max_EDH);
%
```

Replace by average quartz values

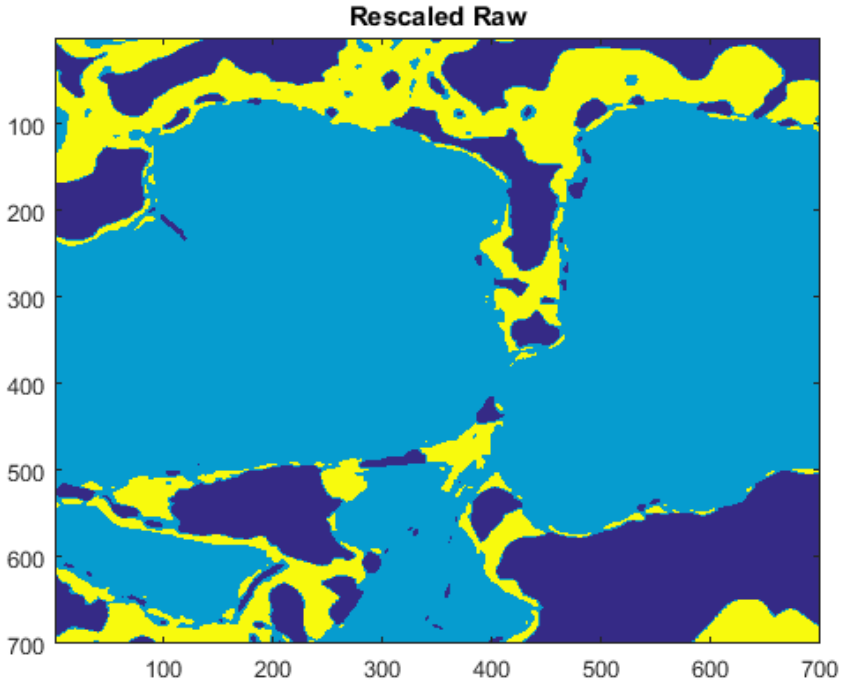
```
if min(M_replace(Ind_rep_E))==min_EDH & max(M_replace(Ind_rep_E))==max_EDH
    M_replace(Ind_rep_E)= avg_Qz;
else
    fprintf('min and max for EDH dont match.....\n')
    return
end

%--------------------------------------------------------------------------
%indexes of gas hydrate pixels
%--------------------------------------------------------------------------
Ind_rep_M = find(M_replace>=min_GH & M_replace<=max_GH);
% replacement by average gas hydrate value
if min(M_replace(Ind_rep_M))== min_GH & max(M_replace(Ind_rep_M))==max_GH
    M_replace(Ind_rep_M)= avg_GH;
else
    fprintf('min and max for methane dont match.....\n')
    return
end

%--------------------------------------------------------------------------
```

1.23. Reshape rescaled array

```
%--------------------------------------------------------------------------
M_replaced = reshape(M_replace,[dim(1), dim(2), dim(3)]);
clear M_replace;
figure; imagesc(M_replaced(:,:,1));
title('Rescaled Raw');
```



1.24. Step 5

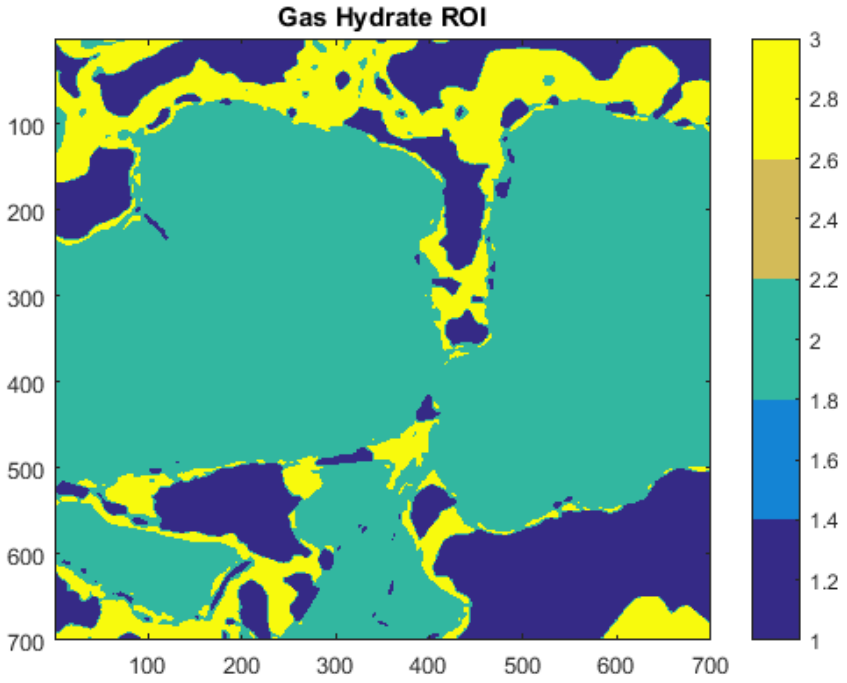
K-means clustering is performed on the rescaled images to obtain segmentation in three classes:

```
clusterS =3;
initialcenters = [avg_li,avg_Qz,avg_GH];
for ii = 1:dim(3)
    R=double(M_replaced(:,:,ii));
```

```
    [r,c,v]=find(R>grenzwert);
    cyl=R>grenzwert;
    R1=cyl.*R;
    [m, n, w]=find(R1);
    G = kmeans(w,clusterS,'Distance','sqeuclidean','start',initialcenters');
    S=sparse(r,c,G,size(R,1),size(R,2));
    M_seg=full(S);
    SegImg(:,:,ii)=M_seg;
    %figure; imagesc(SegImg(:,:,ii)); colormap(parula(5)); colorbar;
end

figure; imagesc(SegImg(:,:,1)); colormap(parula(5)); colorbar;
title('Gas Hydrate ROI');

fclose('all');
```



Published with MATLAB® R2015b