

Détection de la végétation à partir d'une image aérienne multi-spectrale

Saâd Aziz Alaoui, Yassine Jamoud, Samy Haffoudhi

6 mars 2022

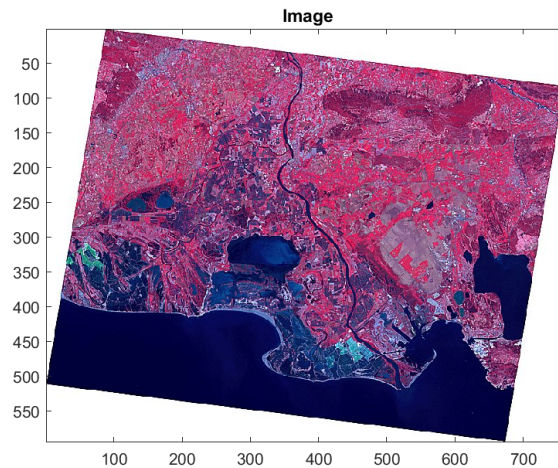
Introduction

On dispose d'une image acquise dans trois bandes de fréquences choisies de sorte à pouvoir identifier les surfaces au sol couvertes par de la végétation. L'objectif du TD est de mettre en œuvre des algorithmes d'analyse de cette image multi-spectrale permettant de détecter la végétation.

1 Réponses aux questions

1. Commençons par ouvrir l'image `camargue.jpg` dans Matlab.

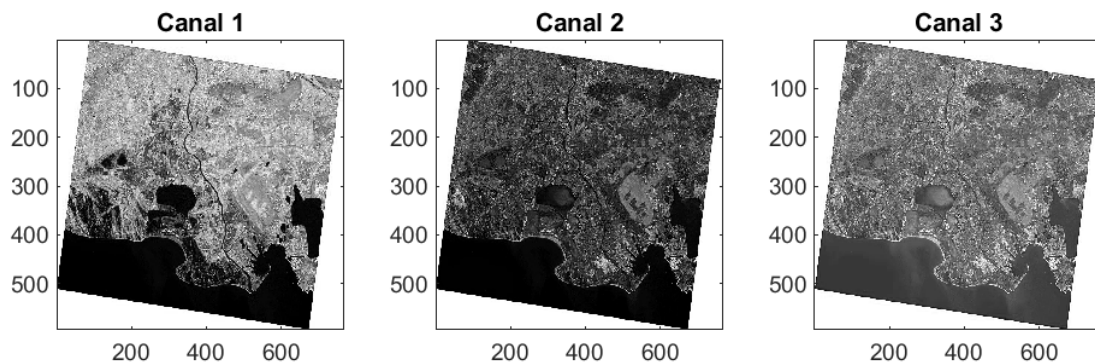
FIGURE 1 – Image



Les couleurs de l'image affichée sont surprenantes. En effet on n'observe par exemple pas de vert alors qu'on cherche à identifier de la végétation. Ces couleurs s'expliquent par le fait que les valeurs des pixels ne correspondent pas à des intensités RVB.

2. Affichons maintenant les trois canaux séparément :

FIGURE 2 – Les 3 canaux de l'image



En nous aidant du spectre de réflectance des végétaux on en déduit les correspondances suivantes :

- Canal 1 \longleftrightarrow Proche infrarouge
- Canal 2 \longleftrightarrow Rouge
- Canal 3 \longleftrightarrow Vert

3. Affichons l'histogramme pour chaque canal :

FIGURE 3 – Canal 1

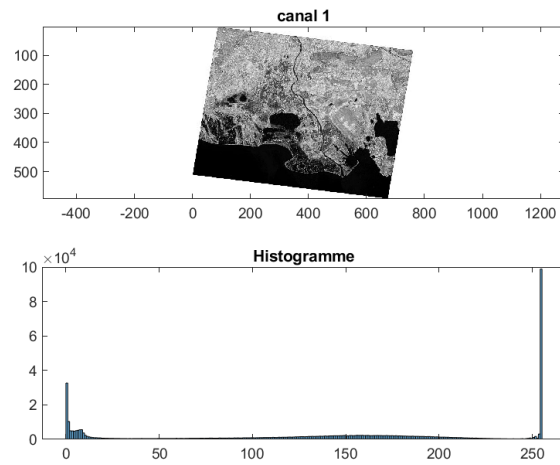


FIGURE 4 – Canal 2

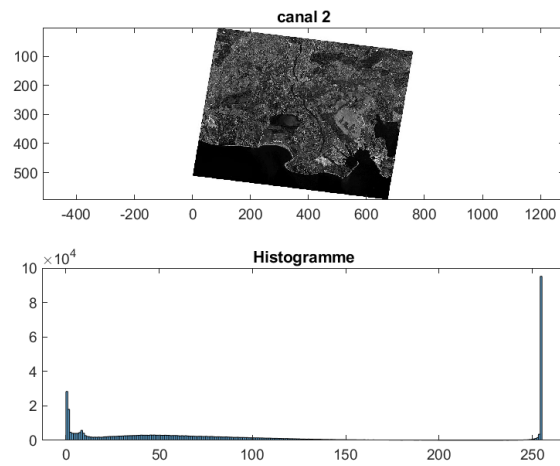
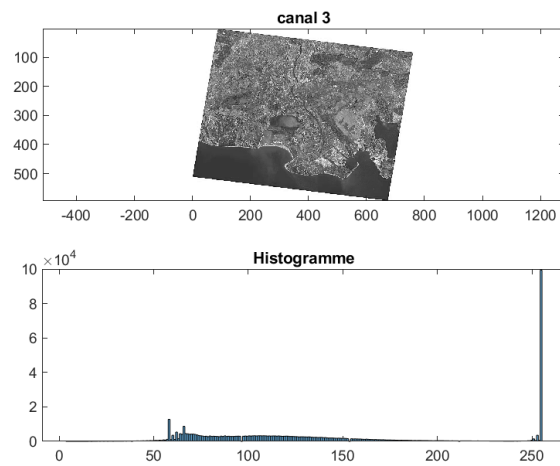


FIGURE 5 – Canal 3



On observe différents "clusters" dans les histogrammes, on peut essayer d'identifier la végétation par un seuillage. On choisit alors un seuil adapté pour chacun des trois canaux (On aurait aussi pu utiliser deux seuils par canal) de sorte à exclure les deux pics, on choisit par exemple respectivement les seuils, 2000, 1000 et 1000.

4. On appliquant le seuillage décrit plus haut, on obtient les masques suivants :

FIGURE 6 – Canal 1

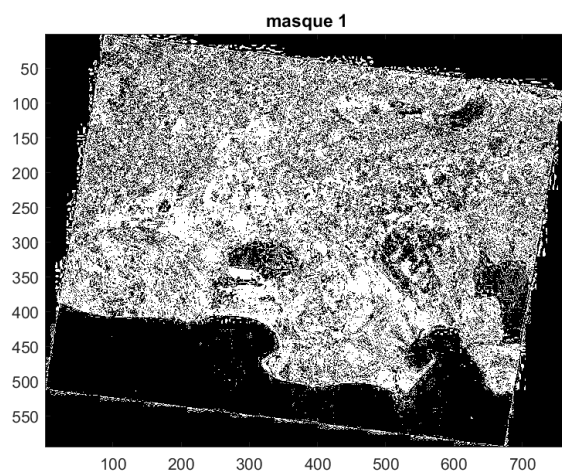


FIGURE 7 – Canal 2

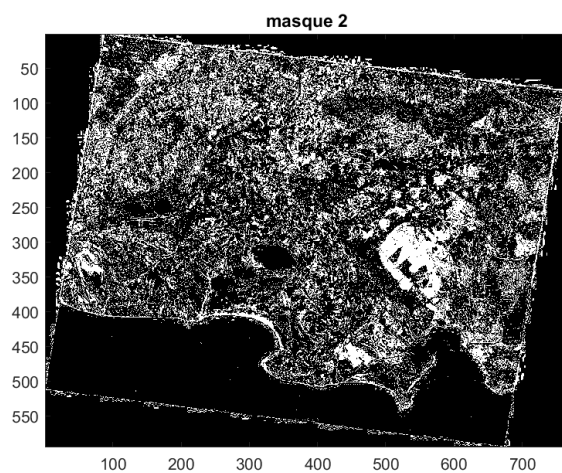
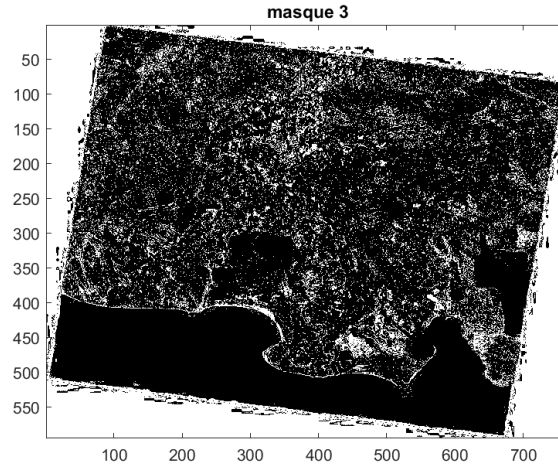


FIGURE 8 – Canal 3



On voit que pour cette première méthode les résultats varient beaucoup d'un canal à l'autre. Le masque 1 est celui sur lequel le plus de végétation serait identifiée tandis que le masque 3 est bien plus sélectif, le masque 2 est entre les deux. On en déduit qu'il faut mettre en place des méthodes plus robustes pour ce problème de détection.

5. Rappel des bandes de longueurs d'onde :

- Bleu : 300 à 500 nm
- Rouge : 600 à 800 nm
- Vert : 500 à 580 nm
- PIR : 0.8 à 2 μm

Ainsi, d'après le graphique, la végétation se distingue le plus des autres éléments dans la bande PIR mais on peut encore le confondre avec le béton.

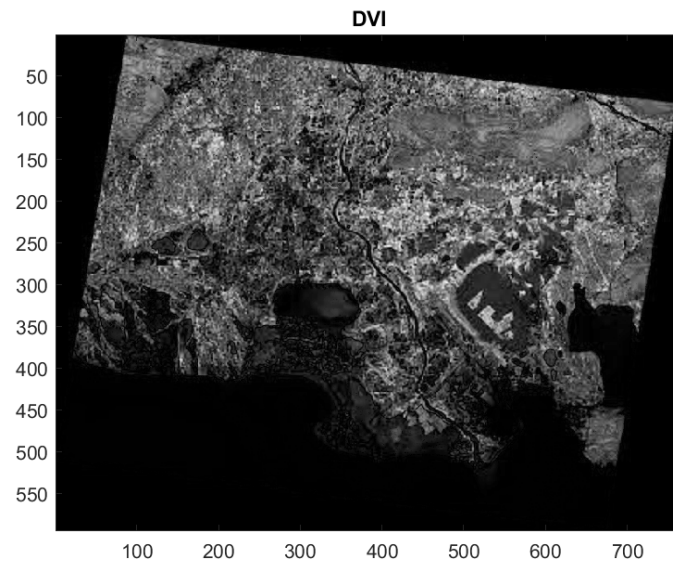
6. Le NDVI est défini par :

$$NDVI = \frac{PIR - R}{PIR + R}$$

Cet indice est adapté à la détection de la végétation car il prend en compte le domaine PIR où la végétation se distingue le plus et lui soustrait le canal R (avant de normaliser) où la végétation émet moins. Par exemple, le béton émet de manière quasiment identique entre ces deux domaines, on pourra alors le distinguer de la végétation à l'aide de l'indice. Par ailleurs, cet indice est aussi utilisé pour estimer l'état de santé de la végétation.

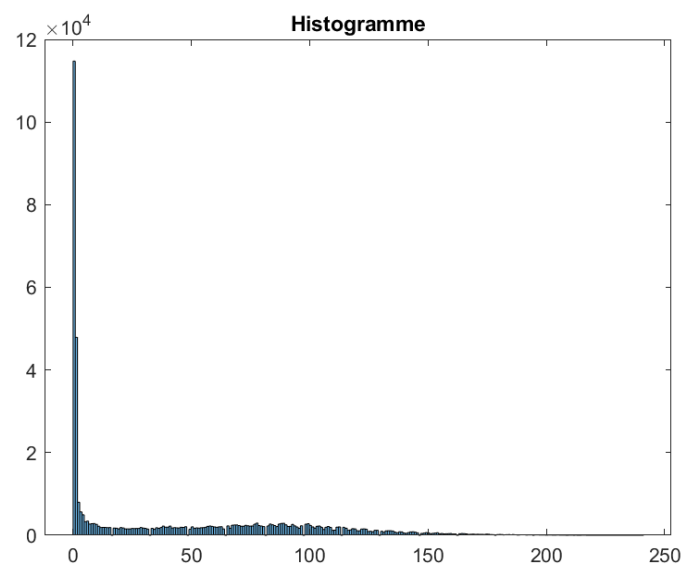
7. Affichons l'image correspondante au DVI :

FIGURE 9 – DVI



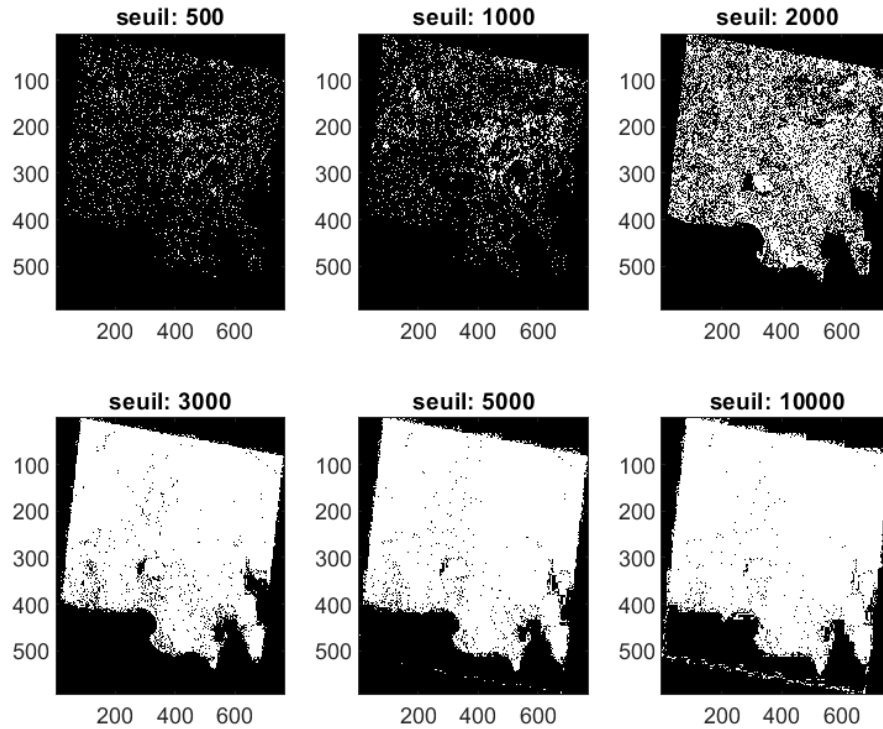
Et l'histogramme correspondant :

FIGURE 10 – Histogramme



Enfin, représentons les masques obtenus pour différentes valeurs de seuil :

FIGURE 11 – Histogramme



On observe alors que :

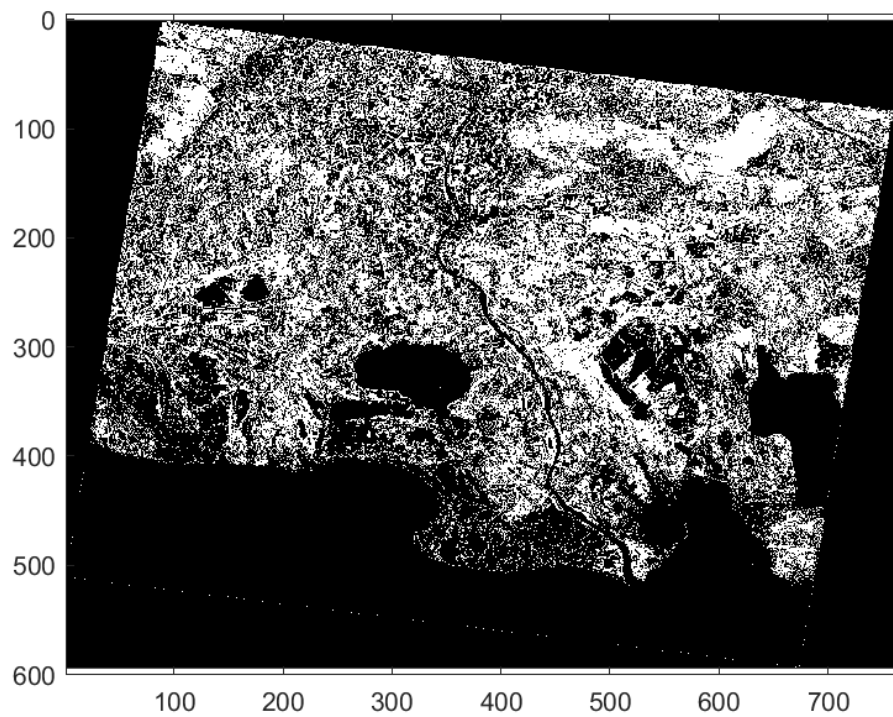
- Pour des valeurs trop faibles du seuil on est trop sélectif
- Pour des valeurs trop élevées du seuil on est trop permissif
- Pour des valeurs entre ces deux extrêmes on obtient des résultats semblant satisfaisant, par exemple avec ici un seuil valant 2000
- Les résultats pour des bons choix du seuil semblent meilleurs que ceux obtenus précédemment par seuillage des différents canaux

8. Enfin, on peut envisager une méthode de détection par le biais de la massification non-supervisée. On choisit d'utiliser la méthode K-means Clustering.

Pour une valeur de K donnée, on détermine alors les K classes et on sélectionne celle correspondant au DVI le plus important.

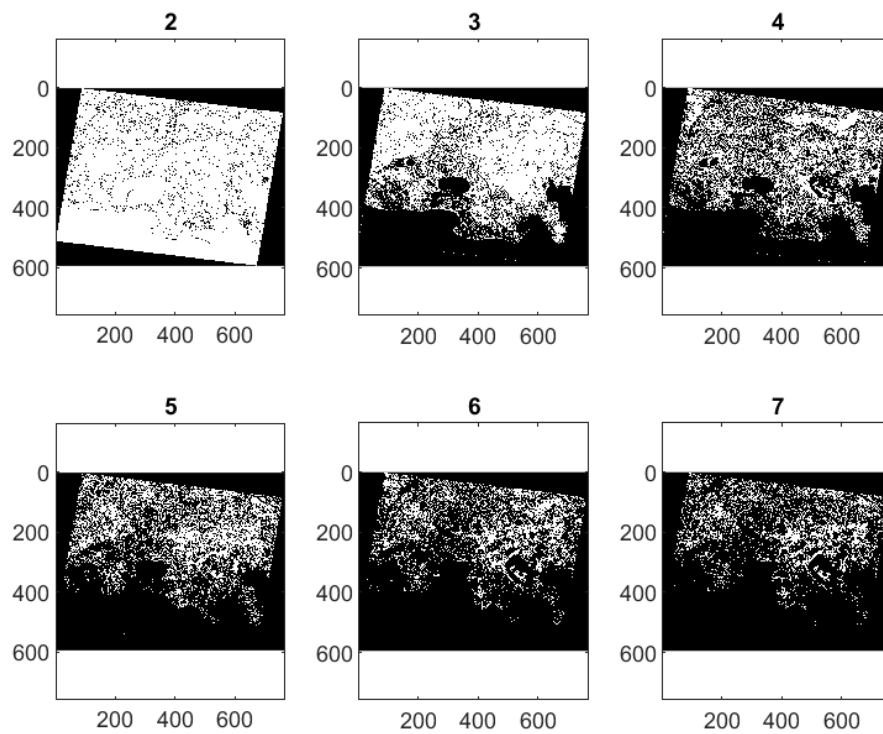
Par exemple, pour $K = 4$, on obtient l'image suivante :

FIGURE 12 – Résultat pour $K = 4$



Pour différentes valeurs de K , on obtient les résultats suivants :

FIGURE 13 – Résultats pour différents K



On observe alors que :

- Pour des valeurs trop faibles de K , c'est-à-dire pour peu de clusters, on obtient pas de bons résultats car le découpage est trop vaste
- À l'inverse, pour une valeur trop importante de K , on découpe l'image en trop de clusters et obtient cette fois trop peu de végétations
- Pour $K = 3$ ou $K = 4$ dans ce cas on obtient de bons résultats

Conclusion

Pour conclure, lors de ce TP, nous avons pu implémenter différentes méthodes pour la détection de la végétation : le seuillage d'histogramme, l'utilisation du NDVI et la classification non supervisée. À partir de l'implémentation de ces méthodes, nous avons en particulier prendre conscience de l'importance du choix des paramètres vis à vis du résultat qui sera retourné, par exemple, les valeurs de seuils ou encore, le nombre de clusters. Il convient de toutes les essayer avec différentes valeurs de paramètres et comparer les résultats pour déterminer celui qui semble meilleur. En effet, ne disposant pas de vérité terrain, nous ne pouvons pas déterminer de manière exacte quelle méthode a fourni les meilleurs résultats.

A Code Matlab

```
1 close all
2 clear all
3
4 image = imread('camargue.jpg');
5 [M,N,C] = size(image);
6 PIR = double(image(:,:,1));
7 R = double(image(:,:,2));
8 V = double(image(:,:,3));
9
10 figure(); imagesc(image); title('Image')
11
12 figure();
13 subplot(1,3,1)
14 imagesc(PIR); colormap gray
15 title('Canal 1')
16 subplot(1,3,2)
17 imagesc(R); colormap gray
18 title('Canal 2')
19 subplot(1,3,3)
20 imagesc(V); colormap gray
21 title('Canal 3')
22
23 for i = 1:3
24     figure()
25     subplot(2,1,1)
26     imagesc(image(:,:,i)); colormap gray; axis equal
27     title(sprintf("canal %d", i))
28     subplot(2,1,2)
29     histogram(image(:,:,i), 256);
30     title('Histogramme')
31 end
32
33 seuils = [2000 2000 1000];
34 for i = 1:3
35     seuil = seuils(i);
36     h = histcounts(image(:,:,i), 256);
37     canal = image(:,:,i);
38     masque = zeros(M,N);
39     ind = find(h(canal + 1) <= seuil); % valeurs passer 1
40     masque(ind) = 255;
41     figure(); imagesc(masque); colormap gray
42     title(sprintf("masque %d", i))
43 end
44
45 % NDVI
46
47 DVI = abs(PIR - R);
48 NDVI = (PIR - R) ./ (PIR + R);
49 figure(); imagesc(abs(DVI)); colormap gray; title('DVI')
50 figure(); imagesc(abs(NDVI)); colormap gray; title('NDVI')
51
52 figure()
53 histogram(DVI, 256);
54 title('Histogramme')
55
56 figure()
57 seuils = [500,1000,2000,3000,5000,10000];
58 for i = 1:length(seuils)
```

```

59     seuil = seuils(i);
60     h = histcounts(DVI, 256);
61     masque = zeros(M,N);
62     ind = find(h(DVI + 1) <= seuil);
63     masque(ind) = 255;
64     subplot(2,3,i)
65     imagesc(masque); colormap gray
66     title(sprintf("seuil: %d", seuil))
67 end
68
69 % K-means
70
71 K = 4;
72
73 [image_labels, C] = imsegkmeans(image,K);
74
75 channels = zeros(size(image,1),size(image, 2),K);
76 figure;
77 for i=1:K
78     channels(:, :, i) = image_labels == i;
79     subplot(1, K, i);
80     imagesc(channels(:, :, i)); axis equal; colormap gray;
81 end
82
83 dvis = zeros(K);
84 DVIs = C(:,1) - C(:,2);
85 [max_dvi, ind] = max(DVIs);
86
87 figure(); imagesc(channels(:, :, ind)); axis equal; colormap gray;
88
89 k_s = 2:7;
90 imgs_opti = zeros(size(image,1),size(image, 2),K);
91
92 for i=1:length(k_s)
93     [image_labels, C] = imsegkmeans(image,k_s(i));
94     for j=1:k_s(i)
95         channels(:, :, j) = image_labels == j;
96     end
97     DVIs = C(:,1) - C(:,2);
98     [max_dvi, ind] = max(DVIs);
99     imgs_opti(:, :, i) = channels(:, :, ind);
100 end
101
102 sub_ind = round(length(k_s)/2);
103 for i=1:length(k_s)
104     subplot(2, sub_ind, i);
105     imagesc(imgs_opti(:, :, i)); axis equal; colormap gray
106     title(sprintf("%d", k_s(i)))
107 end
108
109 % figure;
110 % for i=1:length(k_s)
111 %     subplot(2, sub_ind, i);
112 %     C = labeloverlay(image, imgs_opti(:, :, i));
113 %     imagesc(C); axis equal;
114 %     title(sprintf("%d", k_s(i)))
115 % end

```