Rapport de projet Julien MAZZIA
Scilab

Mission A1 :

```
function A1()
    img=readpbm('Encelade_surface.pbm')
    maxPx = max(img) //On cherche la valeur de nuance de gris maximale dans
l'image
    [y, x] = size(img) //On récupére la taille de la matrice

    for i =1:x
        for j=1:y
            //Si la valeur maximale est trouvé on affiche les coordonnées du pixel
correspondant
            if img(j, i)==maxPx
                then printf("Coordonnées x : %d, Coordonnées y : %d",i,j)
            end
        end
    end
 endfunction
```

Le but de cette mission est de trouver les zones d'atterissage qui sont en fait les pixels avec le plus haut niveau de gris.
Nous avons effectué une recherche du ou des pixels les plus élevés de l'image.
Il a fallut d'abord chercher la valeur maximale puis les dimmensions du tableau.
Ainsi avec une double boucle for on peu parcourir l'ensemble de la matrice à la recherche des pixels avec la valeur maximale et afficher leurs coordonnées x, y.

On obtient en coordonnée x 38 et en coordonnées y 22.

Mission A2 :

```
function A2()
    img=readpbm('Mars_surface.pbm')
    moyenne=mean(img) //On cherche la moyenne des pixels de l'image
    r =moyenne*100/255 //On calcule le pourcentage de méthane dans
l'atmosphére
    disp(r)
 endfunction
```

Le but de cette mission est de calculer le pourcentage de méthane dans l'atmosphére de mars.
Dans un premier il a fallu calculer la moyenne des pixels de l'image pour ensuite le transformer en pourcentage et ainsi obtenir le pourcentage de méthane dans l'atmosphére.

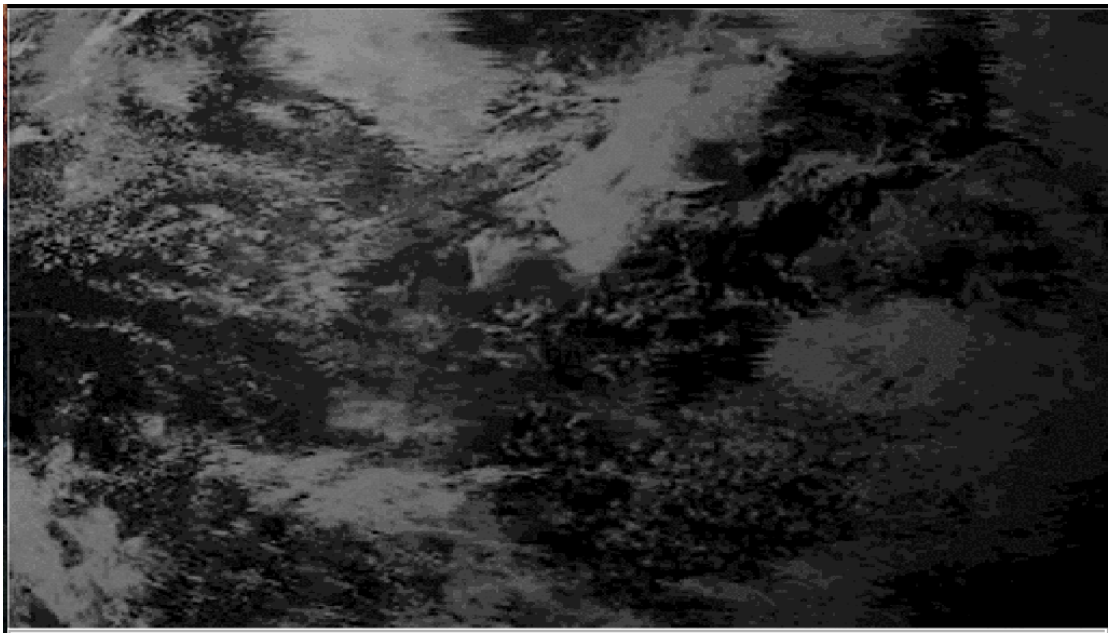
Je trouve comme résultat 54,94% de méthane.

Mission B1 :

```
function B1()
    img=readpbm('Gliese 667Cc_surface.pbm')
    i=max(img) // On cherche la valeur de pixel la plus haute dans l'image
    ratio=255/i //On fait un ratio pour que les valeurs ne dépassent pas 255
    display_gray(img*ratio)
 endfunction
```

La mission B1 consiste à améliorer la luminosité de l'image. Pour cela j'ai calculer un réseau permettant aux pixel de ne pas dépasser la valeur 255 tout en harmonisant l'ensemble.
On avait à la base :

On obtient comme résultat :

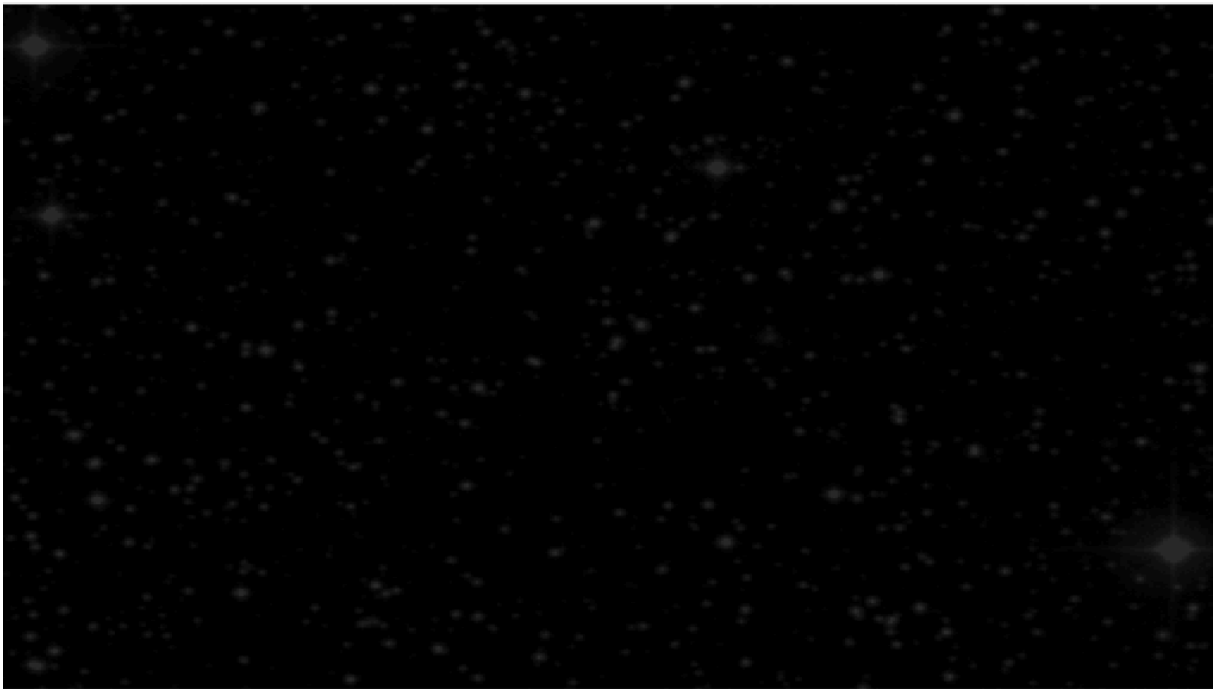


Mission B2 :

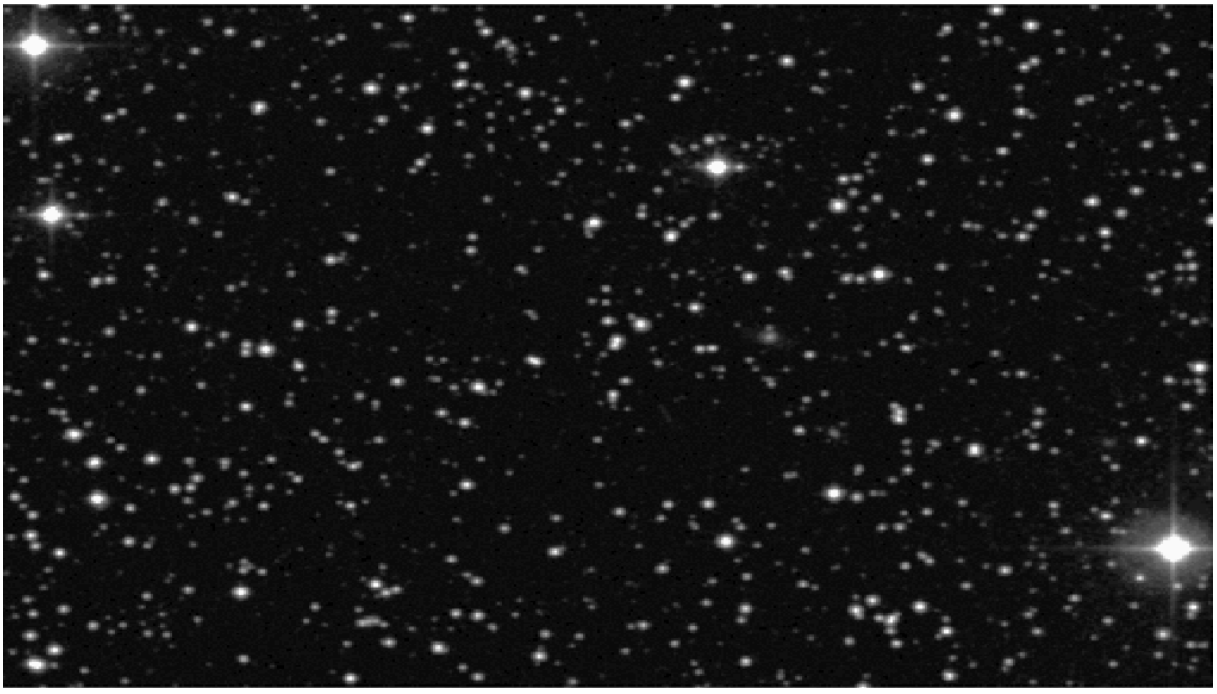
```
function B2()
    img=readpbm('GD61.pbm')
    i=max(img)
    ratio=255/i
    display_gray(img*ratio)
 endfunction
```

La mission B2 a pour but d'amélorier la visibilité de l'image. Pour cela on utilise le même code que la mission précédente.

Avant :



Après :



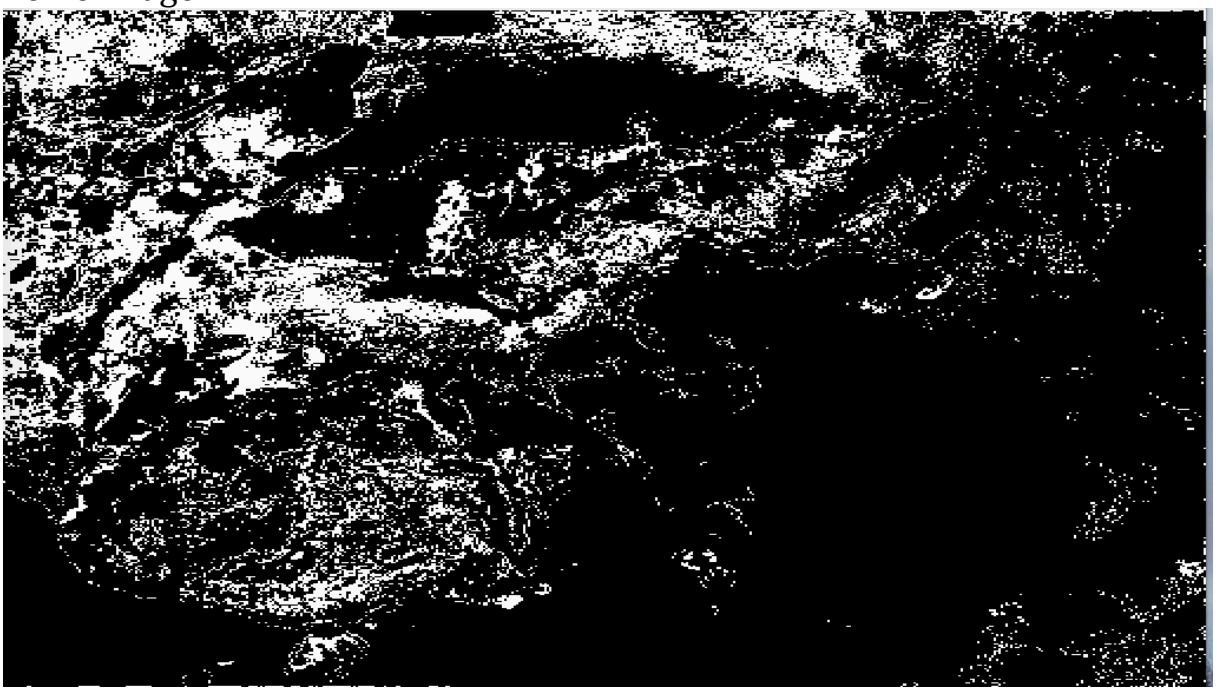
Mission B3 :

```
function B3()
    img=readpbm('HD215497.pbm')
    [y,x] = size(img)
    for i = 1:x
       for j=1:y
            if ((img(j, i)>0) & (img(j, i)<64)) then
               img1(j,i)=255
               img2(j, i)=0
               img3(j, i)=0
               img4(j, i)=0
            elseif ((img(j, i)>64) & (img(j, i)<128)) then
               img1(j,i)=0
               img2(j, i)=255
            elseif ((img(j, i)>128) & (img(j, i)<192)) then
               img2(j, i)=0
               img3(j, i)=255
            elseif ((img(j, i)>192) & (img(j, i)<256)) then
               img3(j, i)=0
               img4(j, i)=255
            end

       end
    end
    display_gray(img1)
    scf
    display_gray(img2)
    scf
    display_gray(img3)
    scf
    display_gray(img4)

 endfunction
```

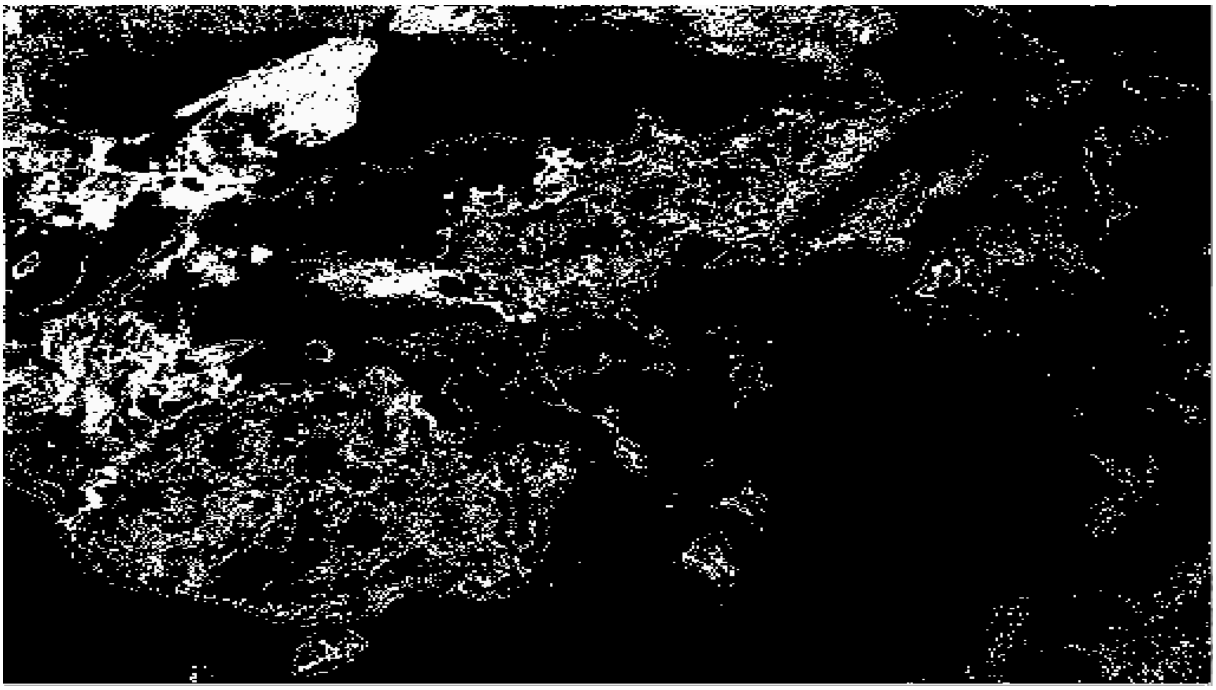
La mission B3 consiste à séparer l'image en 4 zones distinctes.
Pour cet mission je créé 4 matrices dans lesquel je met en évidence un intervalle de bits.
Par exemple dans l'image 1, les bits entre 0 et 64 seront en noir et le reste en blanc.
On a donc 4 images avec un filtrage différent.

Image avec les pixels entre 0 et 64 noirs :

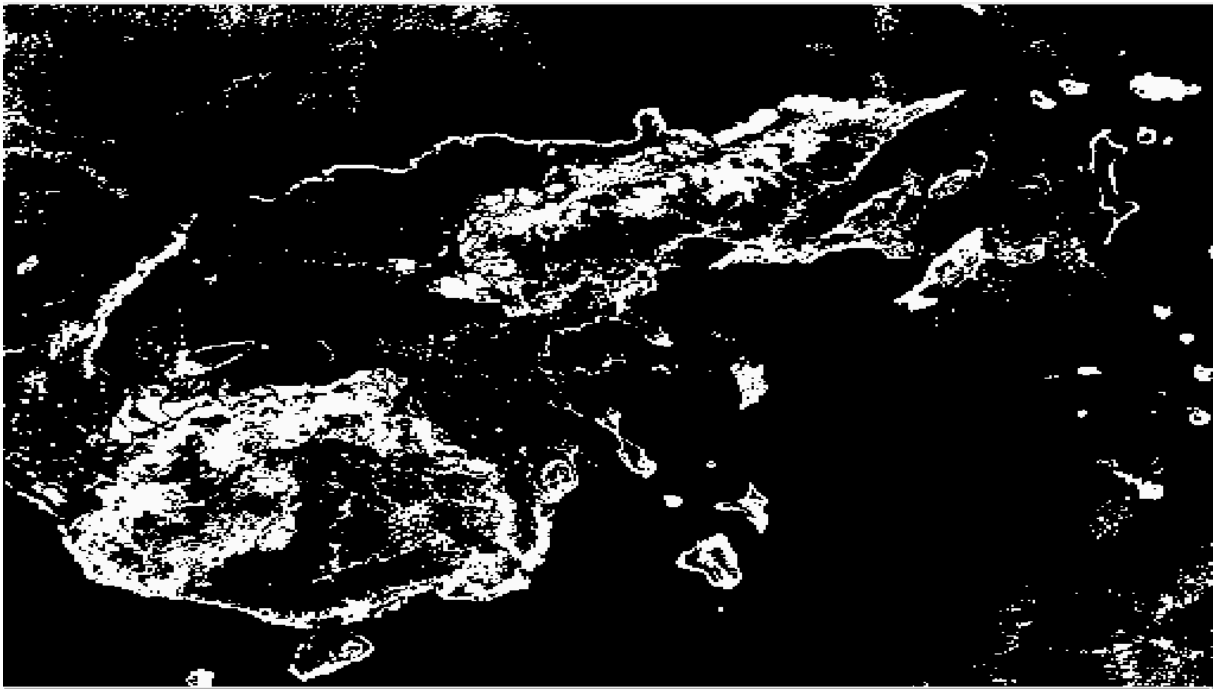
2éme image :



3éme image :



4éme image :



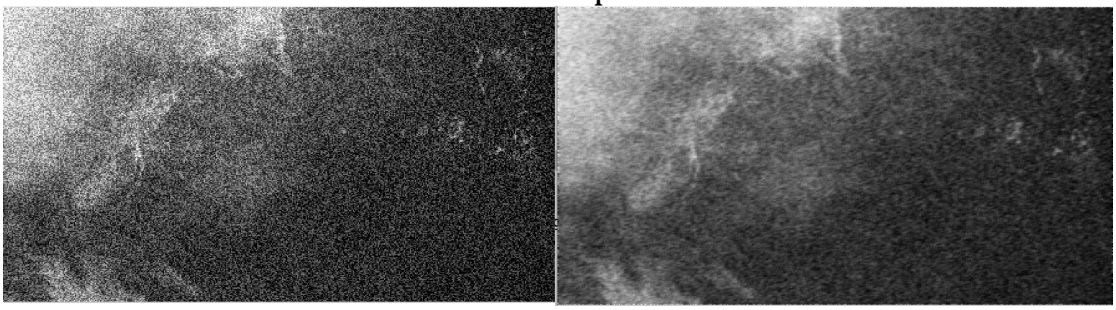
Mission X2 :

```
function X2()
    img=readpbm('Gliese 581d.pbm')

    [y,x] = size(img)
    for i = 2:x-1
```

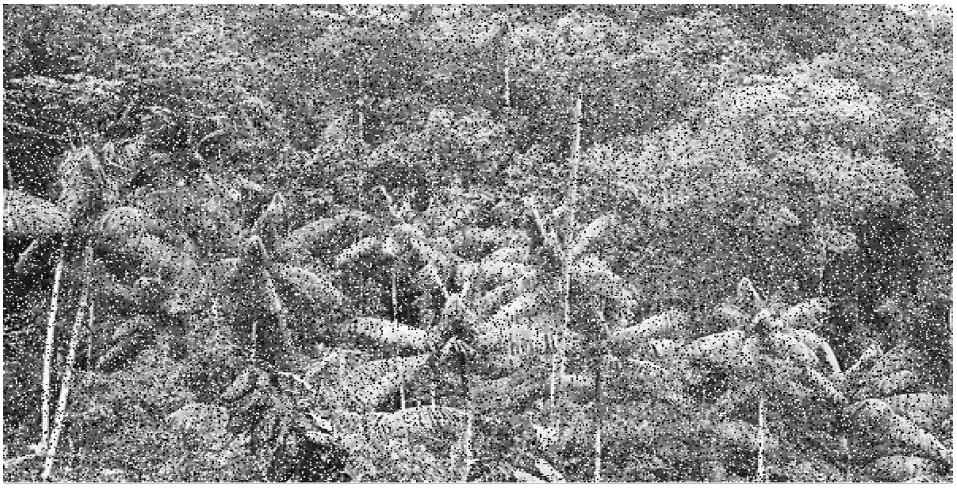
```
        for j=2:y-1
        //On fait la moyenne des différents pixels qui se trouvent autour de celui que l'on regarde
        mean = (img(j-1, i-1)+img(j+1, i+1)+img(j+1, i-1)+img(j-1, i+1)+img(j, i+1)+img(j, i-1)+img(j-1, i)+img(j+1, i))/8
        img(j, i) = mean //On applique cette moyenne au pixel actuel
        end
    end
 display_gray(img)
 endfunction
```

Le but de la mission X2 est de réduire le bruit de l'image.
Le principe est de regarder la nuance de gris des bits aux alentours et d'en faire une moyenne pour remplacer le bit actuel et ansi réduire le bruit de l'image.

Avant : Après :



Avant :



Après :