Phase de développement :

Nous avons cherché dans la phase de développement a suivre les étapes qu’on a décrit dans les schémas bloc de l’étude préliminaire. Pour cela, on a procédé étape par étape, pour etre sur de notre code et comprendre l’ensemble des fonctions, point à point.

Après avoir chargé une image dans scilab grâce a notre fonction, on a pu travailler sur la matrice. Cette fonction permet de convertir l’image en matrice. Il y a également une option qui permet de convertir une image couleur en image en nuances de gris. C’est ce que nous avons utilsé : Nous n’avons travailler qu’avec des images en nuances de gris.

|  |
| --- |
| function **matriceImage**=chargerImage(**path**, **isRGB**)  if **isRGB** == 0 then  **matriceImage** = double(imread(**path**));  else  **matriceImage** = double(rgb2gray(imread(**path**)));  end  endfunction |

Une fois notre image sous forme matricielle on a pu chercher a appliquer notre première étape, le passage d’une image en image fréquentielle. Pour cela, on réalise dans un premier temps une transformée de fourier.

On utilise la fonction de scilab fft. Cette fonction permet de récupérer une matrice de nombres complexes. Ces nombres contiennent donc les informations de modules et phases de notre image. Dans notre cas, on va chercher a appliquer notre tatouage a notre module, et laisser notre phase inchangée, pour pouvoir reconstituer l’image à la fin. Pour cela, nous avons créé une fonction qui permet de calculer la phase et le module :

|  |
| --- |
| ImageFreq = fft(image);  [module,phase] = complexe(ImageFreq);  function [**mod**, **phase**]=complexe(**matrice**)  [N,M] = size(**matrice**)  for x = 1:N  for y = 1:M  **mod**(x,y) = abs(**matrice**(x,y))  **phase**(x,y) = atan(imag(**matrice**(x,y)),real(**matrice**(x,y)))  end  end    endfunction |

Maintenant que nous avons le module de notre image, nous pouvons créer un tatouage pour chercher a l’appliquer. Nous avons choisi un tatouage très basique, qui nous permet de tester nos différentes fonctions. Ce tatouage comporte une symétrie centrale :

|  |
| --- |
| tatouage = [ 1,1,1,1,1,1,1;  1,0,1,1,1,0,1;  1,1,1,1,1,1,1;  0,1,1,1,1,1,0;  1,1,1,1,1,1,1;  1,0,1,1,1,0,1;  1,1,1,1,1,1,1  ] |

La première chose à faire est de dimensionner notre tatouage à la taille de notre image. Le rendu n’est pas forcement très esthétique, il s’agit de 6 gros rectangles noirs. L’objectif qu’on s’est fixé est de mettre en place nos fonctions, sans regarder le côté « tatouage utile ».