Perception, Manipulation et protection d’image

Marie Guénon / Arnaud Tanguy

Tatouage numérique

Table des matières

[Signal pseudo-aléatoire 2](#_Toc379892970)

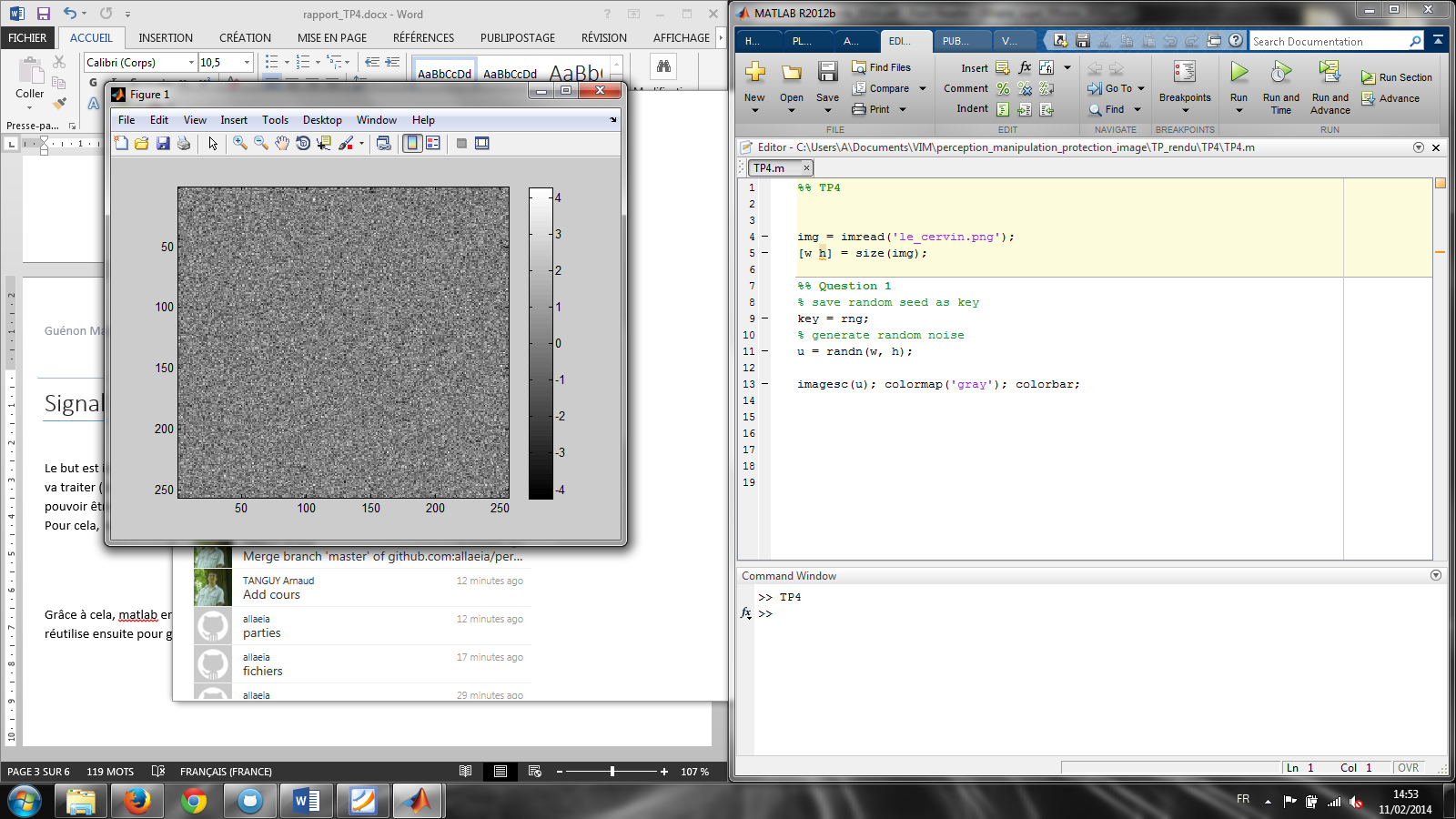
[Génération d’une marque 3](#_Toc379892971)

[Module de décodage 4](#_Toc379892972)

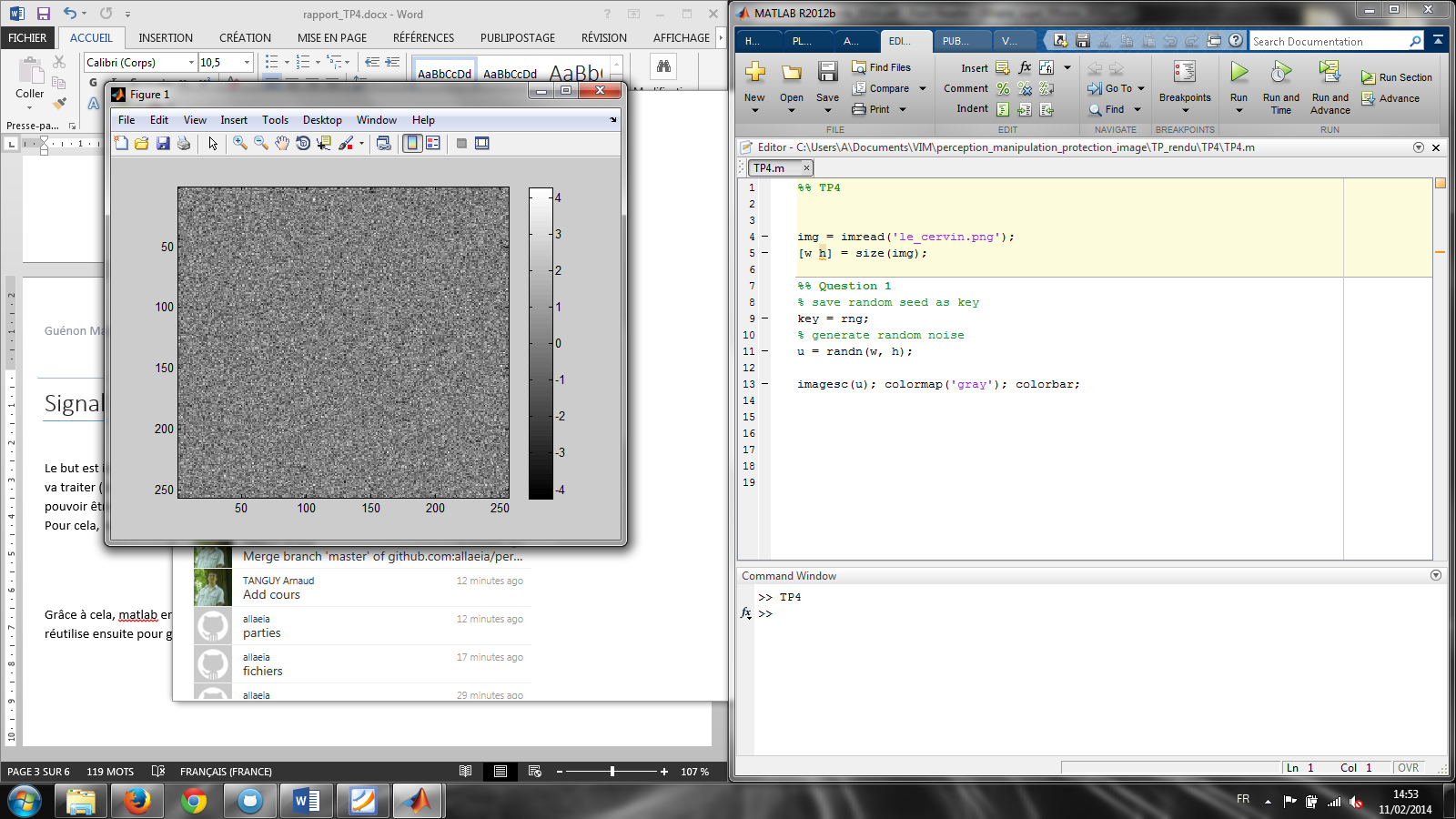
[Décodage 5](#_Toc379892973)

# Signal pseudo-aléatoire

Le but est ici de créer un signal pseudo-aléatoire, c'est-à-dire un bruit blanc, de la taille de l’image que l’on va traiter (ici 256x256) suivant une loi gaussienne centrée réduite . De plus ce bruit blanc doit pouvoir être recréé à partir d’une clé secrète K.  
Pour cela, nous avons utilisé les fonctions de matlab *rng*  et *randn*:



Grâce à cela, matlab enregistre l’initialiseur de l’algorithme de génération (*seed*), et c’est ce chiffre qu’il réutilise ensuite pour générer le bruit blanc. Et on obtient ce genre de résultats :

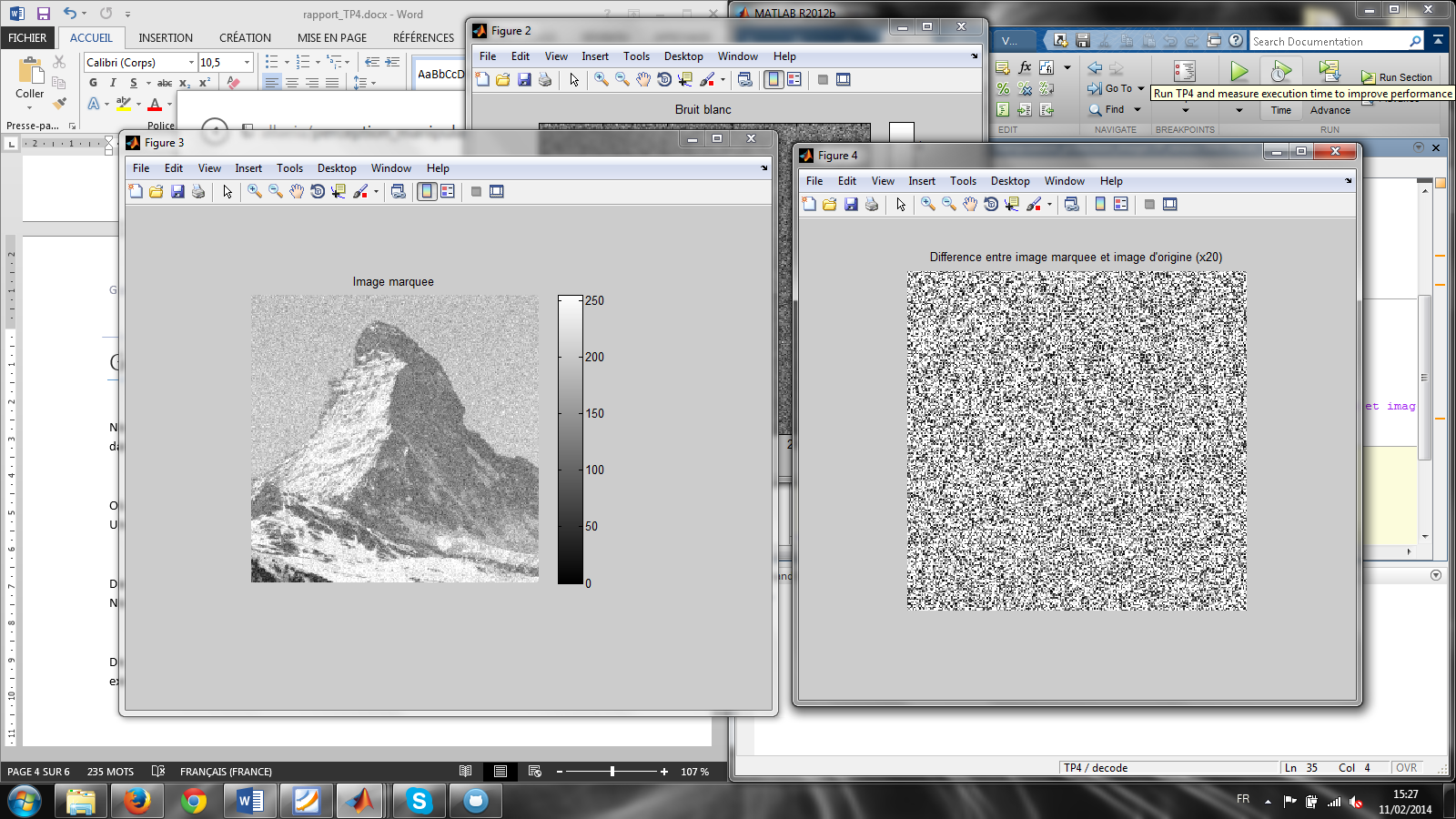


# Génération d’une marque

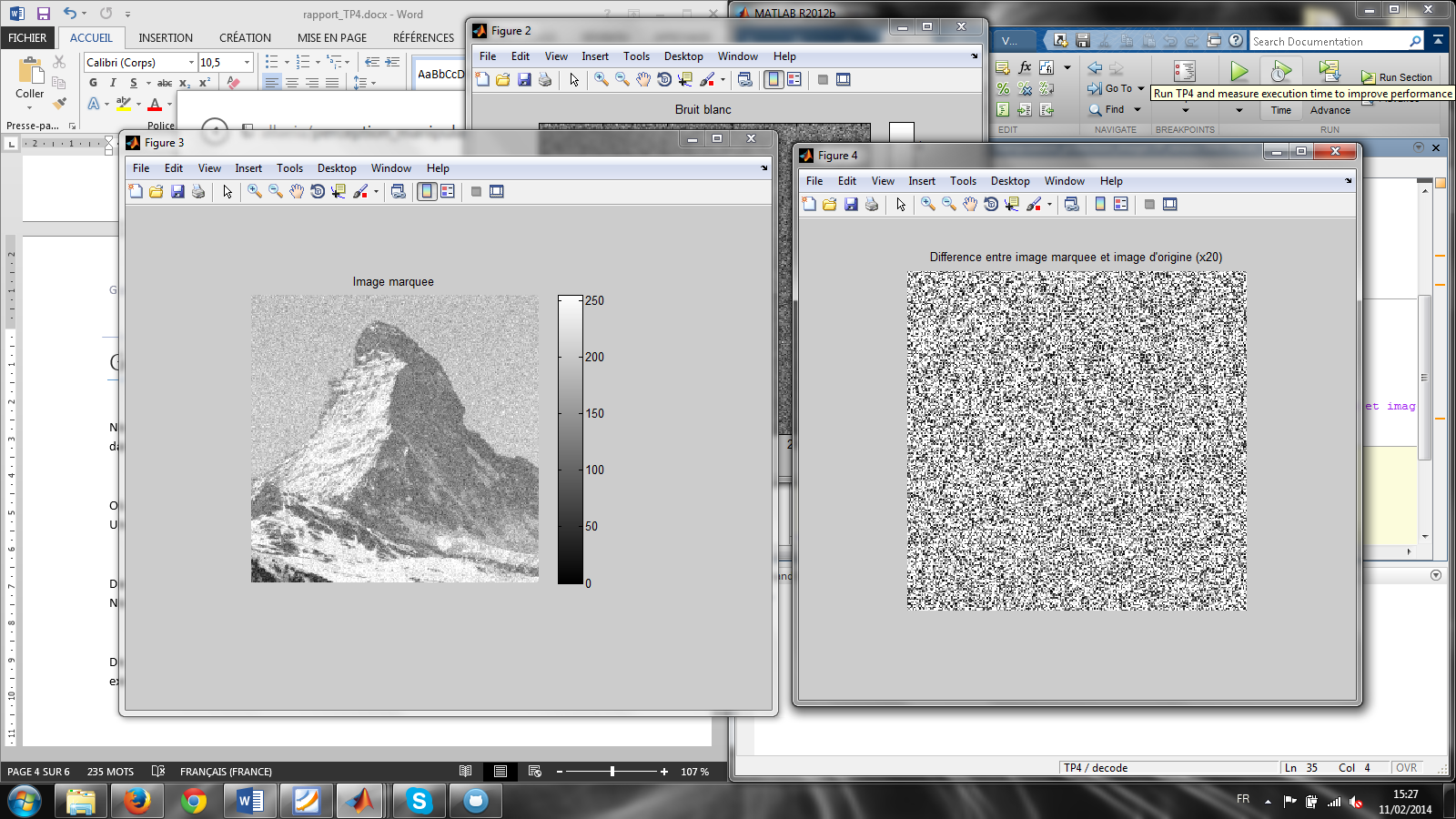
Nous cherchons maintenant à générer une marque ω permettant de dissimuler le bit caché et à l’insérer dans l’image. Pour cela, nous utilisons une modulation du type :

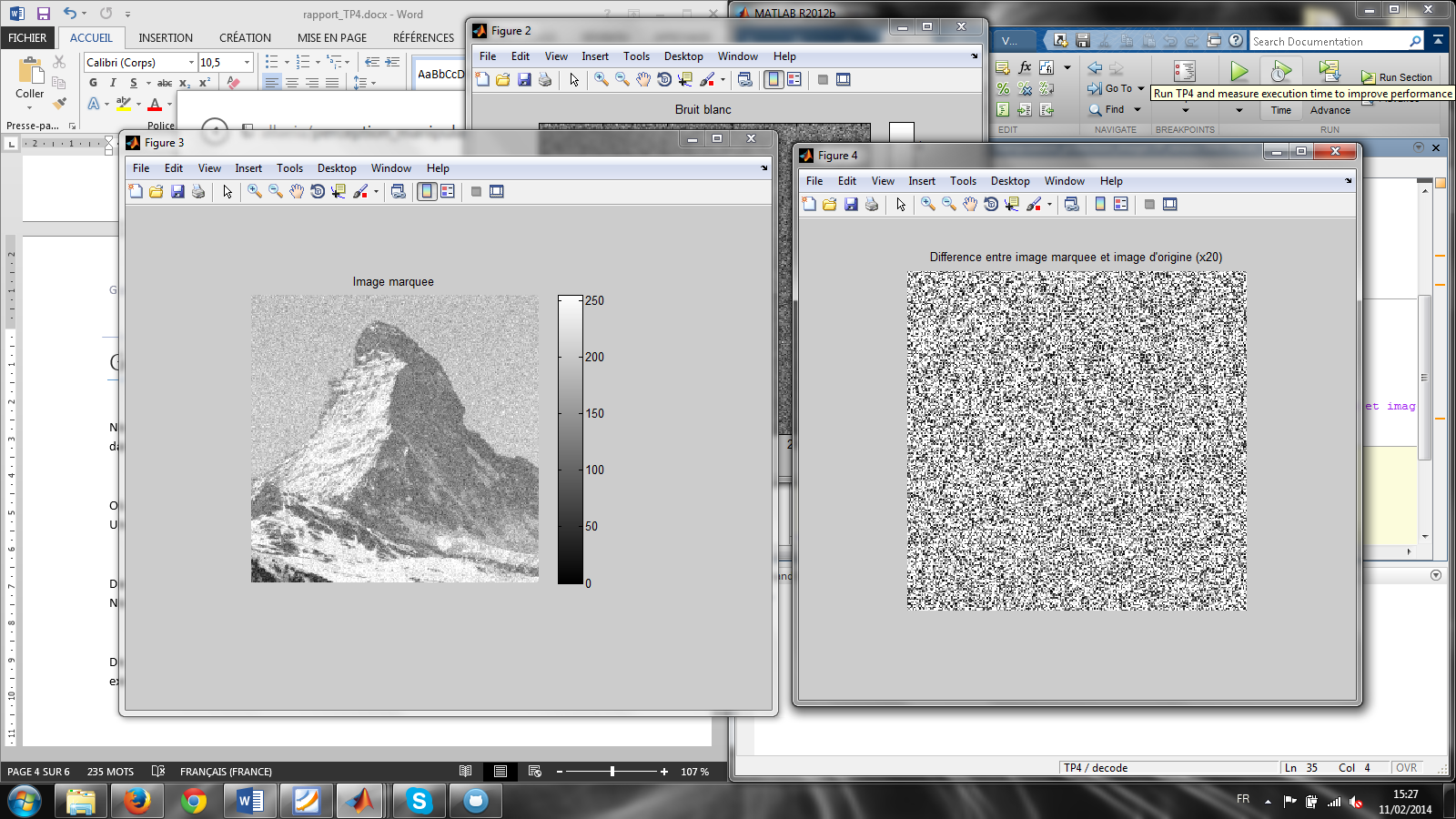
Où *b* est le bit à dissimuler et une constante positive.  
Une fois appliqué à une image *img*, cela nous donne la formule :

Dans notre cas, nous avons pris , *b* le bit que nous voulons cacher dans l’image (ici, nous avons pris *b=0* ) et *u* le bruit blanc que nous avons généré à l’étape précédente. Nous obtenons alors une image du Cervin légèrement bruitée :



De plus, même si ici la différence entre l’image d’origine et l’image marquée n’est pas très visible, celle-ci existe tout de même : (image d’origine à gauche, image marquée à droite et distance entre les deux images en dessous)



# Module de décodage

# Décodage