



FACULTÉ D'ÉLECTRONIQUE ET D'INFORMATIQUE

DÉPARTEMENT D'INFORMATIQUE

Projet de vision par ordinateur

Cacher un texte dans une image

Réalisé par :

- DJEBROUNI Radhia (Groupe: 2)
- ABDELALI Asma Nihad (Groupe: 1)

Responsable du module : L. Abada

Introduction

La stéganographie est un moyen de communiquer secrètement un message, différent de la cryptographie par le fait que le message est caché dans un autre message plus grand de façon qu'il soit indétectable, alors que la cryptographie change chaque caractère par un ou plusieurs autres caractères. Le message conteneur peut être une vidéo, audio, texte ou tout autre format, dans le cas de notre projet le but est de cacher un texte dans une image.

Notre approche pour l'encodage

Nous commençons d'abord par transformer notre message à cacher ainsi que notre image sous format binaire, mais avant il faut s'assurer que les tailles du message et de l'image sont adéquates. Nous plaçons aussi un caractère de fin de message afin de reconnaître la fin du message lors du décodage et d'arrêter le déroulement.

Pour cacher le texte, chaque bit du texte est caché dans un pixel de l'image en sachant que chaque lettre est sur 8 bits. Quant à l'image, chaque pixel est représenté par trois couleurs, r, g et b dont chacune est sur 8 bits. Il faut donc avoir :

Le nombre de caractères du message * 8 < la hauteur * la largeur de l'image

En utilisant l'approche LSB nous cachons 1 bit du texte dans chacune des valeurs RGB d'un pixel avant de passer au suivant. Nous remplaçons le bit de poids faible par un bit du message à envoyer.

En résumé voici les étapes suivies :

- Vérifier la taille du message et de l'image
- Ajouter un caractère de fin de message
- Transformer le message en une chaîne binaire
- Pour chaque pixel de l'image récupérer les valeurs r, g et b
- Remplacer l'LSB de chacune de ces trois valeurs en gardant l'ordre rgb par un bit du message caché

Notre approche pour le décodage

Pour récupérer le message caché dans notre image, nous procédons à l'opération inverse, c'est-à-dire parcourir l'image pixel par pixel, pour chaque pixel nous récupérons ses valeurs RGB nous transformons chacune des valeurs en binaire puis on récupère le bit de poids faible.

Une fois la chaîne en binaire récupérée, nous séparons chaque 8 bits puis les transformons en caractères, à la rencontre du caractère de fin de message précédemment placé on s'arrête.

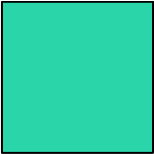
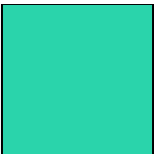
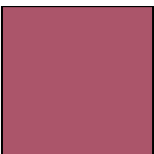
En résumé voici les étapes suivies :

- Pour chaque pixel de l'image récupérer les valeurs r, g et b
- Récupérer l'LSB de chacune de ces valeurs et les concaténer en gardant l'ordre rgb
- Transformer chaque 8 bits du message récupéré en caractère
- S'arrêter au caractère de fin de message

Pourquoi l'Encodage Least Significant Bit (LSB)

Pour l'encodage de type LSB prend en compte le bit de poids faible et le modifie selon le besoin, le choix du bit du poids faible au lieu de celui de poids fort, c'est que le changement apporté est petit. En sachant que nous représentons nos pixels sous un système r, g, b, ces trois valeurs représentent la présence des couleurs rouge, vert et bleu respectivement sur une échelle de 0 à 255, en représentant ces valeurs en binaire le changement de poids faible ne va pas beaucoup influencer la couleur ce qui ne va pas laisser de trace dans l'image conteneur. Mais si nous changeons le bit de poids fort, la couleur va fortement changer.

Exemple représentant l'influence du changement d'un bit sur une couleur :

	Couleur	Intensité de rouge	Intensité de Vert	Intensité de Bleu
Originale	 rgb = (43, 213, 170)	00101011	11010101	10101010
En changeant LSB	 rgb = (42, 212, 171)	00101010	11010100	10101011
En changeant MBS	 rgb = (171, 84, 107)	10101011	01010101	01101010

Nous remarquons qu'au changement du bit de poids fort, la couleur change drastiquement contrairement à l'approche LSB où le changement est imperceptible. Dans une image le résultat aurait été visible.