

Projet Vision par ordinateur
Partie 1 et 2 (2022/2023)

lyes_abada@yahoo.fr

www.eroaya.com/vision

Projet partie N°1: Texte caché dans une image

Une image est représentée sur un nombre fixe de bits (8bits, 16bits...)

Le bit MSB est le bit du poids fort et le bit LSB est le bit du poids faible

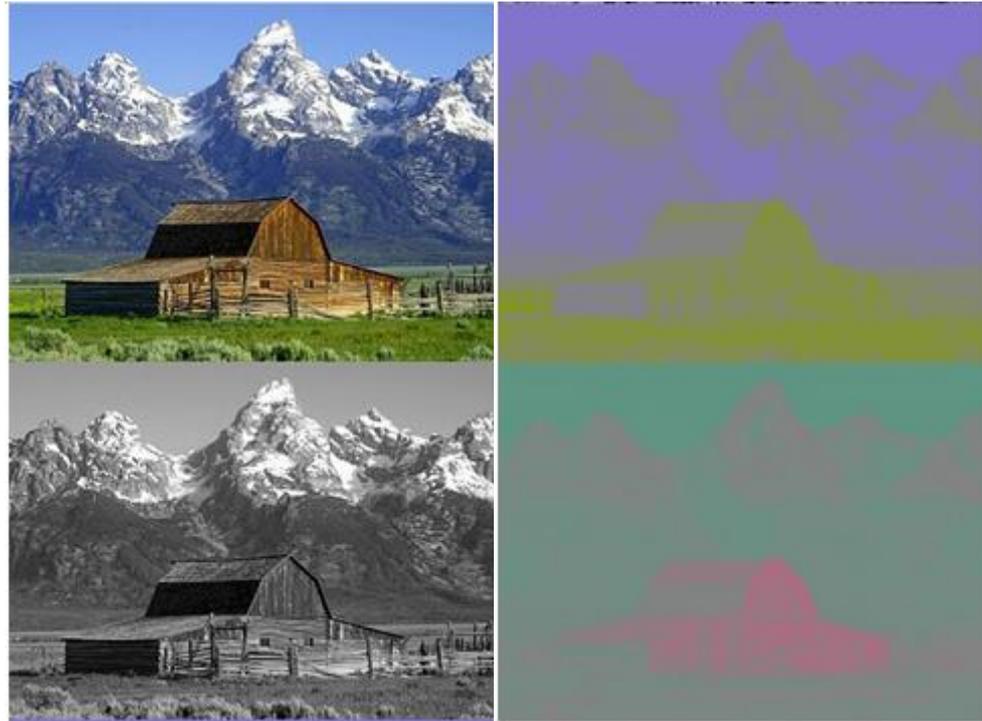
Exemple :

	MSB										LSB					
45473 =	1	0	1	1	0	0	0	1	1	0	1	0	0	0	0	1
45489 =	1	0	1	1	0	0	0	1	1	0	1	1	0	0	0	1
45441 =	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0	1

La couleur

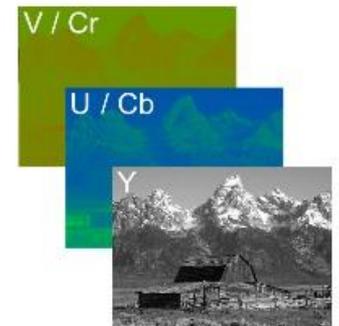
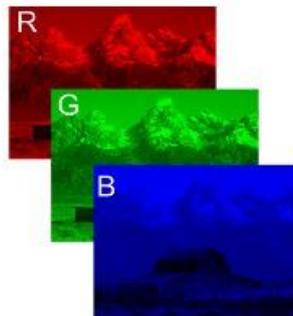
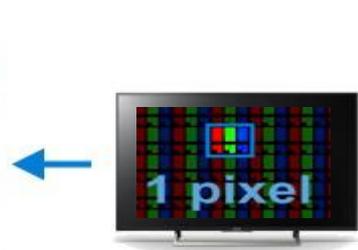
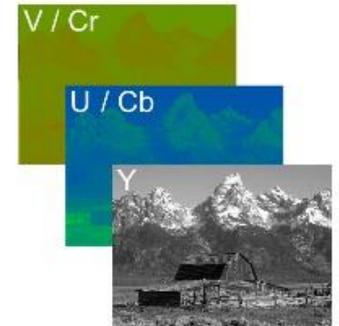
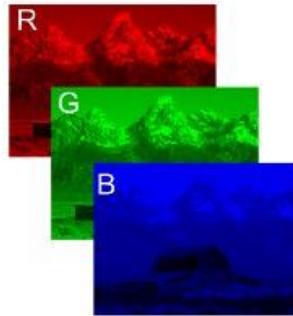
Une image peut être représentée avec plusieurs espaces colorimétriques comme le RGB, le BGR(OpenCV), HSV, YCbCretc.

Dans ce projet, on utilise deux système colorimétriques : le **BGR**(Bleu, Green et Red) et le **YCbCr** qui sépare la luminance et la chrominance



Séparation Luminance (Luma) et Chrominance (Chroma),

La couleur



Projet partie N°1: Texte caché dans une image

Objectif du projet ce projet est de cacher un texte secret dans une image **imgA** en profitant des avantages de l'espace colorimétrique **YCbCr** qui sépare la luminance et la chrominance.

Comme l'œil humain est sensible à la luminance plus qu'à la chrominance, le texte doit être ajouter dans les bits de poids faible de la chrominance (CbCr).

Projet partie N°1: Texte caché dans une image

a) L'émetteur du texte secret :

- Le texte secret doit être converti en une image **imgB** BGR sur 8bit.
- l'image **imgA** doit être converti à l'espace colorimétrique YCbCr sur 16 bits.
- l'image **imgB** doit être insérer dans l'image **imgA** dans les bits de poids faible de **CbCr** tel que :

Une conversion **RGB → YCbCr** ensuite **YCbCr → RGB** peut engendrer un taux d'erreur égal à **± 8** sur 16bits (0 → 65535)

- l'image doit être par la suite convertir et sauvegarder en RGB sur 16bits,

b) Le récepteur du texte secret :

- Le récepteur doit faire l'opération inverse pour afficher l'image **imgB** qui contient le texte secret.

Projet partie N°1: Texte caché dans une image

L'émetteur du texte secret



Message : USTHB...

imgB

USTHB...

010011010101...

imgA



Le récepteur du texte secret

imgB

USTHB...

010011010101...

imgA



Projet partie N°1: Texte caché dans une image

Remarques :

L'application doit contenir deux fonctionnalités séparées :

- l' émetteur qui consiste à l'insertion du texte dans une image puis la sauvegarder
- Le récepteur qui consiste à la lecture et l'affiche de l'image imgB à partir d'une image.

Projet partie N°1: Texte caché dans une image

Remarques :

- Le travail se fera en trinômes,
- La programmation se fera avec python. Il est fortement recommandé d'utiliser (OpenCV, Python, VSCode), sinon, si vous utilisez d'autres environnements, vous devez remettre un programme sans aucune dépendance.
- le code source et le rapport doivent être compressés dans un seul fichier portant les noms des étudiants
- Envoyez le fichier à l'adresse suivante : abada.lyes@gmail.com.
- Le dernier délai pour envoyer le rapport et le code est **le 03/11/2022**
- La démonstration se fera **le 07 et le 08 Novembre 2022**.

Vous devez mentionner les configurations nécessaires pour exécuter le code dans le rapport (bibliothèques à téléchargées...)

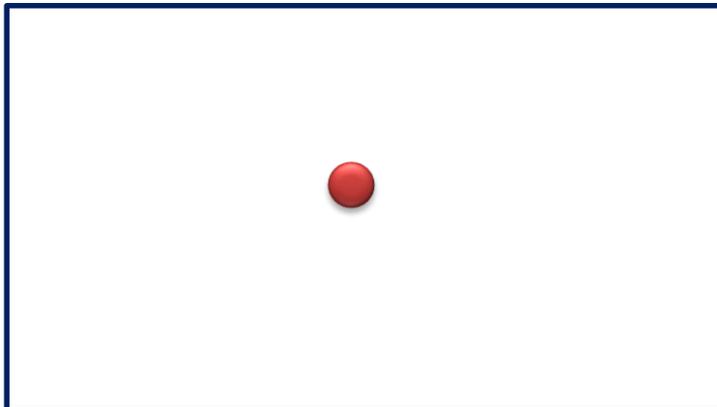
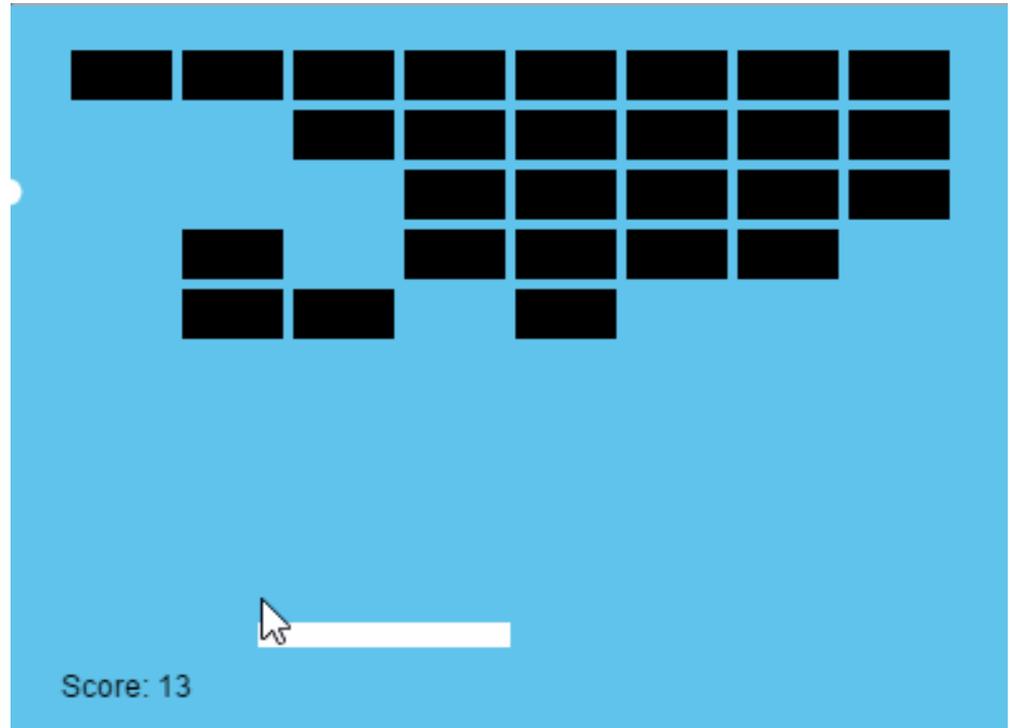
Projet partie N°2: Bricks Breaker Game

- Dans cette partie, il est demandé de créer un petit jeu « Bricks breaker ».
- Il est demandé de déplacer la raquette inférieure pour frapper la balle qui rebondit sur les murs.
- le déplacement doit être effectué en utilisant la caméra en déplaçant une couleur.
- La position de la couleur dans la frame définit la position de la raquette dans notre jeu.

Ecrire un programme qui permet de détecter la position d'une couleur (de votre choix) par la caméra en utilisant OpenCV.

Pour faciliter la tâche mettez la couleur dans une feuille blanche (pour garantir un fond blanc).

Projet partie N°2: Bricks Breaker Game



Projet partie N°2: Bricks Breaker Game

Remarques :

- Proposer au moins deux améliorations de votre choix pour la première partie, par exemple :
 - Ajouter des boutons pour la configuration automatique de la couleur.
 - Ajouter des niveaux de difficulté,
 - Utiliser un API de détection d'objets pour déplacer la raquette avec les gestes
 - ...etc.
-
- Rédiger un rapport sur le travail effectué (entre 4 et 5 pages).

Projet partie N°2: Bricks Breaker Game

Remarques :

- La programmation se fera avec Python. Il est fortement recommandé d'utiliser VSCode, sinon, si vous utilisez d'autres environnements, vous devez remettre un programme sans aucune dépendance.
- Seulement les fonctions vues en salle de TP sont autorisées pour réaliser le projet sauf la partie amélioration.
- le code source et le rapport doivent être compressés dans un seul fichier portant les noms des étudiants
- Envoyez le fichier à l'adresse suivante : abada.lyes@gmail.com.
- Le dernier délai pour envoyer le rapport et le code est **le 10/12/2022**
- La démonstration se fera **le 12 et le 13 décembre 2022**.

Vous devez mentionner les configurations nécessaires pour exécuter le code dans le rapport (bibliothèques à téléchargées...)