

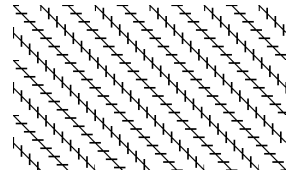
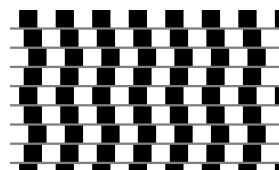
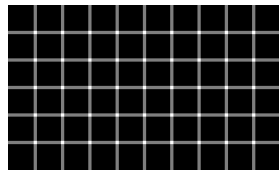
TP - Illusions d'optique

Objectif(s)

- ★ Manipuler des images
- ★ Renforcer les boucles imbriquées

Exercices obligatoires

Nous allons produire plusieurs images permettant de créer des illusions d'optiques.
Un aperçu des illusions ciblées est présenté ci-dessous :



Exercice 1 – Dégradé

Une première illusion d'optique consiste à dessiner un rectangle gris monochrome sur un fond dégradé.

Question 1

Écrivez une fonction `rectangle_gris(struct imagePPM img)` qui prend en paramètre une image et dessine un rectangle gris en son milieu. Le rectangle sera centré, occupant $\frac{2}{3}$ de la largeur de l'image, et 30 pixels d'épaisseur. On utilisera la couleur médiane (127, 127, 127). et le rectangle occupera un espace en largeur de $\frac{1}{6}$ à $\frac{5}{6}$. Testez sur une image classique (par exemple de Fleur) et observez.

Question 2

Écrivez une fonction `degrade` qui crée une image de dimensions 300×500 , avec le niveau max de couleur à 255, et la rempli d'un dégradé de gris (de (255, 255, 255) à (0, 0, 0)).

Question 3

Incrustez le rectangle monochrome sur le dégradé, et observez le résultat surprenant.

Exercice 2 – Points ?

Question 1

Écrivez une fonction `eclaircir_ligne(struct imagePPM img, int ligne)` qui ajoute 128 à chaque composante de couleur d'une ligne horizontale de l'image, sans dépasser 255. Écrivez de même une fonction `eclaircir_colonne(struct imagePPM img, int col)` qui éclaircit une colonne.

Question 2

En utilisant les fonctions précédentes, écrivez une fonction qui crée une image noire de dimension 360×590 , puis éclaircit des lignes et colonnes pour laisser des carrés de 50 pixels de côté. L'épaisseur des zones claires sera de 10 pixels.

On peut améliorer l'illusion en remplaçant les intersection par un disque blanc de 6 pixels de rayons, ce que vous pourrez faire en renforcements.

Exercice 3 – Courbes droites

Question 1

Écrivez une fonction `ligne_de_carres(struct imagePPM img, int ligne, int decalage)` qui prend en paramètre une image, un numéro de ligne et un décalage, et dessine une ligne horizontale d'épaisseur 40 pixels en alternant des carrés 40×40 noirs et blancs. La ligne la plus haute de la ligne de carré est celle passée en paramètre, et le décalage en paramètre indique le pixel de début du premier carré noir. Vous penserez à ne pas utiliser de coordonnées qui sortent du cadre de l'image.

Question 2

Écrivez une fonction `ligne_grise(struct imagePPM img, int l)` qui trace sur une image une ligne horizontale de couleur gris moyen, d'épaisseur 5 pixels et centrée en `l`.

Question 3

Complétez l'image de l'illusion en alternant lignes de carrés et ligne grises. Vous utiliserez des décalages successifs utilisant les valeurs `[20, 30, 40, 30, 20, 30, ...]`.

Exercice 4 – Ouvrons le placard

Pour cet exercice, vous utiliserez l'image `placard.ppm`. On pourra travailler par effet de bord sur l'image.

Question 1

On souhaite ouvrir le placard. Pour cela, vous prendrez la porte de gauche (des pixels `(71, 4)` à `(131, 186)`) en la décalant de 64 pixels vers la gauche. Vous pourrez remplir l'espace libéré avec du noir, ou avec l'image `fantome.ppm`.

Question 2

Même question en entrouvrant la porte de droite (des pixels `(135, 4)` à `(196, 186)`) en la décalant de seulement 32 pixels vers la droite (attention il peut y avoir un piège!).

Exercice 5 – Niveaux de couleur

Pour refaire les niveaux d'une image, on peut utiliser la méthode suivante :

- pour chaque couleur, on cherche la valeur minimum v_m et la valeur maximum v_M apparaissant parmi les pixels de l'image, on obtient la gamme de cette couleur : l'intervalle $[v_m, v_M]$
- puis, on réaffecte à chaque pixel une nouvelle valeur v_N sur cette couleur, calculée en projetant l'intervalle de départ sur l'ensemble des valeurs disponibles $[0, 255]$. On obtient cette valeur à partir de la valeur initiale v_I en calculant

$$v_N = \frac{255(v_I - v_m)}{v_M - v_m}$$

Question 1

Écrivez une fonction `minimum_r(img)` qui calcule la valeur minimale des coefficients de rouge sur toute l'image. Écrivez de même une fonction `maximum_r(img)`. Modifiez les fonctions pour qu'une fonction renvoie les minimum et maximum de chaque couleur.

Question 2

Écrivez le programme qui refait les niveaux de couleur de l'image avec la méthode précédente. Vous pourrez tester votre programme en révélant l'image `hell.ppm`.

Renforcements

Exercice 6 – Illusion de Zolner

Question 1

Pour cette illusion, nous vous laissons un peu plus l'autonomie, en vous appuyant sur l'illustration. Toutes les lignes obliques sont inclinées de 45° (un pixel à droite pour un pixel vers le bas), elle sont espacées de 50 pixels, les

lignes traversantes font 21 pixels de long et sont espacés de 15 pixels. Commencez par faire des lignes d'un seul pixel d'épaisseur.

Question 2

Améliorez votre illusion en donnant une épaisseur de trois pixels à la ligne.