

1 Transformations d'images

1.1 Buts

- Création d'un programme complet.
- Utilisation de tableaux multi-dimensionnel.
- Lecture et écriture dans un fichier.
- Gestion d'erreurs.

1.2 Énoncé

Dans ce travail pratique, il s'agit, à partir d'une image donnée, d'effectuer:

1. Trois symétries.
2. Un effet de photomaton qui reproduit l'image quatre fois sans perte d'information.
3. Un zoom de l'image d'une puissance de 2 fois (2 fois, 4 fois, 8 fois, ...).

Être capable de lire et d'écrire une image en niveaux de gris au format `.pgm`.

1.2.1 Travail supplémentaire

Lorsque ces trois étapes sont terminées, effectuer une rotation de l'image d'un angle demandé à l'utilisateur à la ligne de commande.

1.3 Cahier des charges

Un fichier dans lequel se trouve stockée une image en niveaux de gris en format `.pgm` est fournie par l'utilisateur et son nom est entré à la ligne de commande. Cette image devra être stockée dans une matrice (un tableau de taille dynamique bi-dimensionnel).

Il faudra ensuite créer les fonctions permettant d'effectuer:

- les symétries horizontales, verticales, et centrales.
- le photomaton.
- le zoom.
- et en bonus la rotation.

Finalement, le résultat devra être sauvegardé sous le format `.pgm` en niveaux de gris sous un nom entré par l'utilisateur.

1.3.1 Le photomaton

Une étape du photomaton consiste à reproduire l'image donnée 4 fois en 4 fois plus petite. Cette opération sera effectuée plusieurs fois. A chaque étape il n'y a pas de perte d'information: chaque pixel se retrouve simplement déplacé (voir fig. 1).

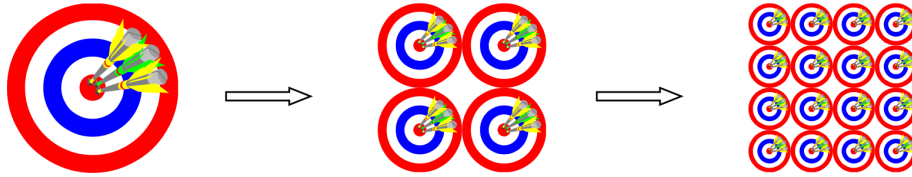


Figure 1: Exemple d'application de l'algorithme du photomaton où à la fin on a 4 fois l'image originale en 4 fois plus petit.

Pour une image de dimension 8×8 , la transformation peut s'observer sur la fig. 2

1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8		1,1	1,3	1,5	1,7	1,2	1,4	1,6	1,8
2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8		3,1	3,3	3,5	3,7	3,2	3,4	3,6	3,8
3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8		5,1	5,3	5,5	5,7	5,2	5,4	5,6	5,8
4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8		7,1	7,3	7,5	7,7	7,2	7,4	7,6	7,8
5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	→	2,1	2,3	2,5	2,7	2,2	2,4	2,6	2,8
6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8		4,1	4,3	4,5	4,7	4,2	4,4	4,6	4,8
7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8		6,1	6,3	6,5	6,7	6,2	6,4	6,6	6,8
8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8		8,1	8,3	8,5	8,7	8,2	8,4	8,6	8,8

Figure 2: La position avant (gauche) et après (droite) la transformation du photomaton.

A partir de cet exemple généraliser cette transformation pour chaque groupe de 4 pixels. Pour bien comprendre la transformation, il faut examiner ce qui se passe sur la première ligne.

1.3.2 La rotations

Soit un angle α et un point $P = (x, y)$. Le point $P' = (x', y')$, après rotation de α autour de l'origine, est obtenu à l'aide la formule

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \cos \alpha - y \sin \alpha \\ x \sin \alpha + y \cos \alpha \end{pmatrix}.$$

Après application de la rotation, vous observerez probablement que l'image est criblée de petits trous (les pixels après rotation ne tombent pas exactement sur des positions entières). Trouvez un moyen d'y remédier.

1.4 Remarques

Sur le site du cours, vous trouverez le fichier `lena.pgm` qui contient une image à traiter. Cette image est carrée et de dimension 256×256 . Elle contient des

pixels stockés sous forme d'entiers, sur 256 niveaux de gris (de 0 à 255).

Vous pouvez visualiser l'image à l'aide de la commande `display lena.pgm`.

Le format `pgm` est un format texte qui est formaté de la façon suivante:

1. Le texte `P2` sur la première ligne du fichier indiquant que l'image est en niveaux de gris.
2. Des lignes de commentaires commençant par le caractère `#`.
3. Les dimensions de l'image.
4. Le nombre maximal de niveaux de gris.
5. Les valeurs des pixels.

Pour vous aider, vous trouverez également un code sur `cyberlearn`, dans le fichier `negatif.rs` qui:

1. Charge un fichier image au format `.pgm` entré par l'utilisateur à la ligne de commande;
2. Convertit le fichier en un vecteur d'entiers;
3. Calcule le négatif de l'image;
4. Sauve l'image dans le répertoire courant sous le nom `foo.pgm`.