Transformation d'images par applications quasi-affines

Maxime PETITJEAN
Guillaume HAYETTE

Professeur encadrant : Mr David COEURJOLLY

Sommaire

- Introduction
- Description de l’algorithme
  - Application quasi-affine
  - Les pavés
  - L’algorithme de transformation
  - La périodicité
- Déroulement du TER
  - Problèmes rencontrés
  - Démonstration
- Conclusion

Introduction

- Le sujet
  - Obtenir un algorithme optimisé de transformation d’image à partir de deux documents

- Travail à effectuer
  - Analyser ces documents expliquant l’algorithme
  - Coder l’algorithme

Applications quasi-affines

- Une application quasi-affine est une application discrète définie par :

$$x' = \left\lfloor \frac{\mathrm{ax} + \mathrm{by} + \mathrm{e}}{\omega} \right\rfloor$$
$$y' = \left\lfloor \frac{\mathrm{cx} + \mathrm{dy} + \mathrm{f}}{\omega} \right\rfloor$$

- On la définit par un matrice, et un vecteur :

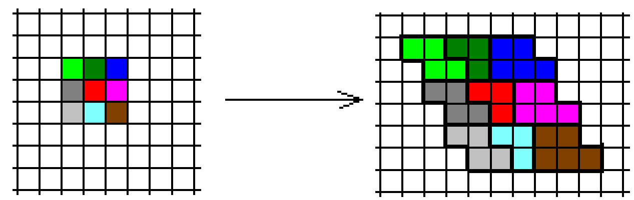
$$A = \frac{1}{\omega}\begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad V = \begin{pmatrix} e & f \end{pmatrix}$$

Applications quasi-affines

- Contractante
  - L'image finale est plus petite que l'image initiale
- Dilatante
  - L'image finale est plus grande que l'image initiale
- Inversible
  - L'application inverse permet de revenir sur l'image initiale

Les pavés

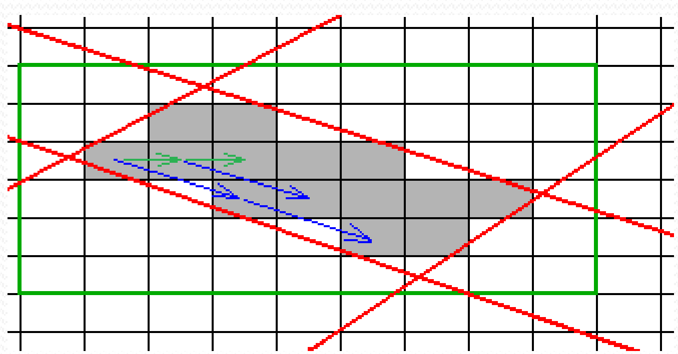
- Pour chaque pixel de l'image initiale, nous obtenons un ensemble de pixels dans l'image finale



- Ce pavé est borné par quatre droites obtenues à partir de l'application elle-même

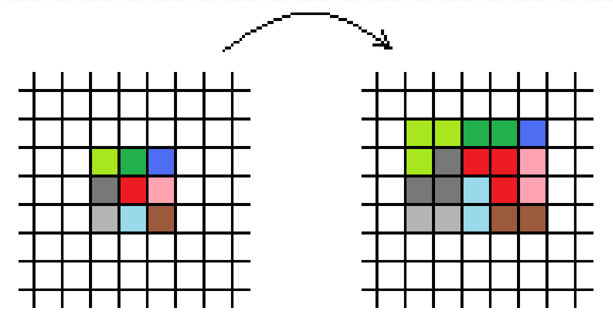
$$x' = \left\lfloor \frac{ax + by + e}{\omega} \right\rfloor \longrightarrow \omega x' \leq ax + by + e < \omega\,(x' + 1)$$

$$y' = \left\lfloor \frac{cx + dy + f}{\omega} \right\rfloor \longrightarrow \omega y' \leq cx + dy + f < \omega\,(y' + 1)$$



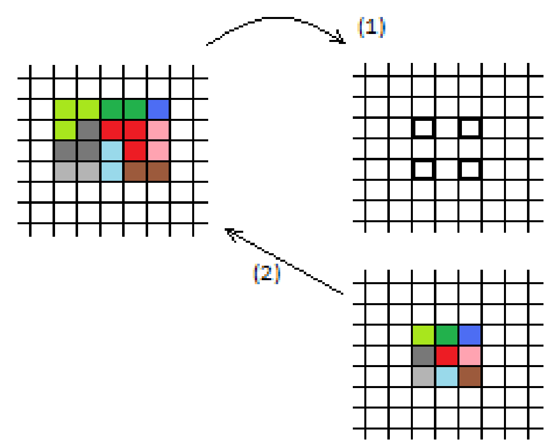
Algorithme de transformation

- Pour une application dilatante
  - Pour chaque point (i, j) de l’image initiale
    - Utiliser l’application inverse pour calculer $P_{i,\,j}$
    - Assigner à chaque point de $P_{i,\,j}$ la couleur de (i, j)



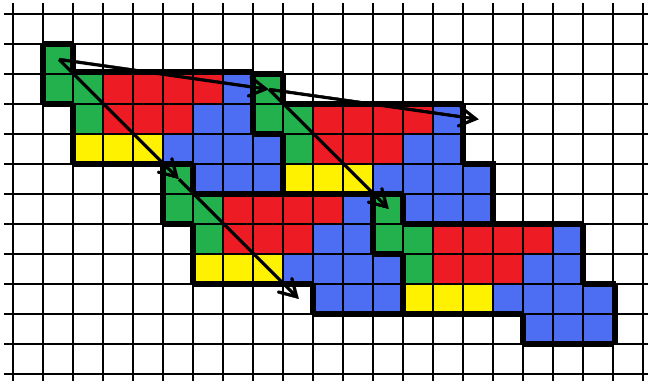
Algorithme de transformation

- Pour une application contractante
  - Calculer les coins de l'image finale
  - Pour chaque point (i, j) de l'image initiale
    - Utiliser l'application pour calculer $P_{i,\, j}$
    - Assigner à (i, j) une couleur dépendant des couleurs de l'ensemble des point de $P_{i,\, j}$



Périodicité

- Répétition des pavés de manière périodique



- Il existe un pavé fini I_0 et deux vecteurs, tels que l'image peut être remplie par translation du pavé I_0 par les deux vecteurs

Algorithme

Données : Image initiale I, application quasi-affine f

si f contractante

alors

F = application quasi-affine associée à f

Calculer les quatre sommets de l'image finale

min_i, max_i = abscisse minimale et maximale de ces sommets

min_j, max_j = ordonnées minimale et maximale de ces sommets

Sinon, f est dilatante

F = application quasi-affin inverse de f (notée f^{-1})

min_i, max_i = abscisse minimale et maximale de l'image initiale

min_j, max_j = ordonnées minimale et maximale de l'image initiale

Finsi

Algorithme

Déterminer l'ensemble P des pavés distinct

Pour i de min_i à max_i faire

Pour j de min_j à max_j faire

Déterminer i', j' et le vecteur V tels que :

- $P_{i,j}$ soit l'image de la translation de $P_{i', j'}$ par le vecteur V
- $P_{i',j'}$ appartienne à P

Si f contractante

Affecter à (i, j) une couleur dépendant des couleurs des points de $P_{i,j}$

Sinon

Affecter aux points de $P_{i,j}$ la couleur du point (i, j)

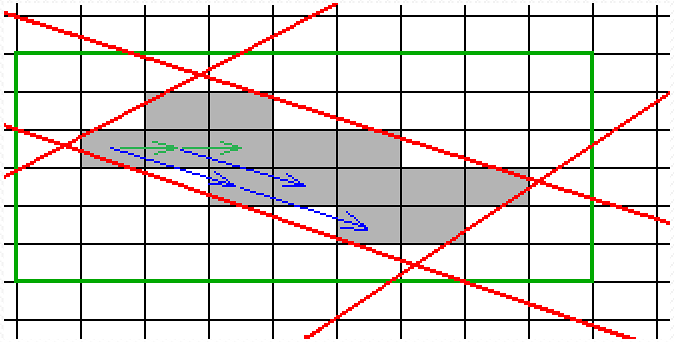
Déroulement du TER

- Lecture et analyse des documents
- Programmation des outils de base
- Codage de l'algorithme

Problèmes rencontrés

- Problème : calcul de variables erroné (C_k et D_k)

- Solution : précision de contrainte sur le calcul de v_o v_1
  - Calculés à partir de $c'v_0 + d'v_1 = 1$
  - Tels que $p = av_0 + bv_1 < \delta_1$



Problèmes rencontrés

- Au niveau programmation
  - Le compilateur arrondit à 4 octets la taille d'une structure
  - Modulo du C dans les négatifs

Démonstration

- Transformation d'une image par l'application :

$$A = \frac{1}{4}\begin{pmatrix} -3 & -4 \\ 4 & -3 \end{pmatrix} \quad V = \begin{pmatrix} 0 & 0 \end{pmatrix}$$

Conclusion

- Découverte de la transformation d'images
- Première expérience dans le monde de la recherche