RDF - TP 1

François Lepan
Benjamin Van Ryseghem

30 janvier 2013

1 Code Scilab

1.1 Analysez le code contenu dans ce fichier et expliquez comment les doubles sommes nécessaires au calcul des moments géométriques sont implantées. Quel est l'intérêt de cette technique ?

Les doubles sommes sont implantées en utilisant les propriétés des calculs matriciels afin d'éviter les boucles d'itérations. L'intérêt de cette technique est qu'étant donné que les calculs matriciels sont très optimisés, il y a un gain de performance (et de lisibilité).

2 Moments d'une forme

2.1 Calcule des valeurs propres et les vecteurs propres de la matrice d'inertie des 4 rectangles

2.1.1 Rectangle Horizontal

Valeurs Propres

(80 1360)

Vecteurs Propres

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Axe Principale

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Moments Principaux

(80 1360)

2.1.2 Rectangle Vertical

Valeurs Propres

(80 1360)

Vecteurs Propres

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Axe Principale

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Moments Principaux

(80 1360)

2.1.3 Rectangle Diagonal

Valeurs Propres

(59 1298)

Vecteurs Propres

$$\begin{pmatrix} -0.7071068 & -0.7071068 \\ -0.7071068 & 0.7071068 \end{pmatrix}$$

Axe Principale

$$\begin{pmatrix} -0.7071068 \\ 0.7071068 \end{pmatrix}$$

Moments Principaux

(1298 59)

2.1.4 Rectangle Diagonal Lissé

Valeurs Propres

(99.673034 1393.7427)

Vecteurs Propres

$$\begin{pmatrix} -0.7080350 & -0.7061774 \\ -0.7061774 & 0.7080350 \end{pmatrix}$$

Axe Principale

$$\begin{pmatrix} -0.7061774 \\ 0.7080350 \end{pmatrix}$$

Moments Principaux

(99.673034 1393.7427)

2.2 Quelle est la différence entre les deux images d'un rectangle diagonal ?

Entre les deux rectangles, même si l'axe principal reste a peu de choses près le même, l'orientation est différente.

2.3 Comment cela influence t'il le calcul des moments ?

De ce fait, les moments sont eux aussi inversés. À cause de cela, deux figures proches ont des moments principaux vraiment différents.

2.4 Calcule des moments principaux d'inertie des différents carrés (6, 10, 30°, 45°, 20)

```
m = inertiaMatrix(image_6);
momentums(m)
        [ 105. 105. ]

m1 = inertiaMatrix(image_10);
momentums(m1)
        [ 825. 825. ]

m2 = inertiaMatrix(image_30d);
momentums(m2)
        [ 842.42024 843.28148 ]

m3 = inertiaMatrix(image_45d);
momentums(m3)
        [ 841.51713 838.53593 ]

m4 = inertiaMatrix(image_20);
momentums(m4)
        [ 13300. 13300. ]
```

Ces moments d'inerties peuvent être utilisés afin de caractériser une forme pour une taille donnée, sans tenir compte de son orientation.

3 Moments normalisés

3.1 Calcul des moments principaux d'inertie en diagonalisant la matrice d'inertie calculée à partir des moments centrés normalisés plutôt qu'à partir des moments centrés

Fonction rdfMomentCentreNormalise : η

```
//carres
m = inertiaMatrixCentered(image_carre6);
momentums(m)
        [ 0.0810185 0.0810185 ]

m1 = inertiaMatrixCentered(image_carre10);
momentums(m1)
        [ 0.0825 0.0825 ]

m2 = inertiaMatrixCentered(image_carre30d);
momentums(m2)
```

```
        [ 0.0840244 0.0841103 ]

m3 = inertiaMatrixCentered(image_carre45d);
momentums(m3)
        [ 0.0854334 0.0851307 ]

m4 = inertiaMatrixCentered(image_carre20);
momentums(m4)
        [ 0.083125 0.083125 ]

// rectangles
m5 = inertiaMatrixCentered(image_rectangle_horizontal);
momentums(m5)
        [ 0.3320313 0.0195313 ]

//on retrouve ici l'inversion des valeurs propres
m6 = inertiaMatrixCentered(image_rectangle_vertical);
momentums(m6)
        [ 0.0195313 0.3320313 ]

m7 = inertiaMatrixCentered(image_rectangle_diagonal);
momentums(m7)
        [ 0.3858502 0.0175386 ]

//on retrouve ici l'inversion des valeurs propres
m8 = inertiaMatrixCentered(image_rectangle_diagnonal_lisse);
momentums(m8)
        [ 0.0239599 0.3350345 ]

//triangles
m9 = inertiaMatrixCentered(image_triangle10);
momentums(m9)
        [ 0.3320313 0.0195313 ]

//on retrouve ici l'inversion des valeurs propres
m10 = inertiaMatrixCentered(image_triangle15d);
momentums(m10)
        [ 0.0195313 0.3320313 ]

m11 = inertiaMatrixCentered(image_triangle45d);
momentums(m11)
        [ 0.3858502 0.0175386 ]

//on retrouve ici l'inversion des valeurs propres
m12 = inertiaMatrixCentered(image_triangle60d);
momentums(m12)
        [ 0.0239599 0.3350345 ]
```

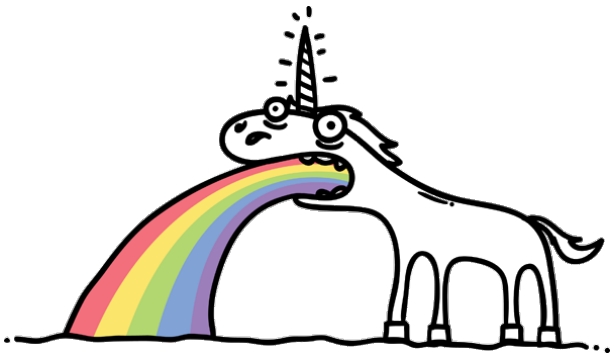
Les moments d'inerties centrées normalisés peuvent servir d'attribut de forme puisqu'il permet de détecter une forme indépendamment de sa taille, de sa position ou de son orientation.

4 Moments invariants

4.1 Calcule des attributs des formes contenues dans les images d'une même forme pour différentes orientations et différentes échelles (les carrés)

```
Hu5(image)
        0.

Hu5(image1)
        0.

Hu5(image2)
        1.378D-14.

Hu5(image3)
        6.501D-15.

Hu5(image4)
        0.
```

Grâce aux moments invariants, on peut comparer des formes similaires, indépendamment de la taille, de la position où de l'orientation des formes. En effet, tous ces carrées retournent une valeur similaire. Il reste plus lourd à calculer que les moments centrés normalisés mais est aussi plus précis.



François Lepan
Benjamin Van Ryseghem