TP CBIR

Nicolas Sidère – 2017

Outils

Pour ce TP, vous pourrez utiliser Matlab, OpenCV, python, what else ?

La recherche d'images par le contenu

La recherche d'images par le contenu est un domaine de recherche très actif depuis plusieurs années maintenant. Avec l'explosion de l'imagerie numérique, nous nous retrouvons souvent avec d'énormes bases d'images et la recherche d'une image particulière est un problème difficile à résoudre. Aujourd'hui, la solution la plus souvent retenue est d'annoter manuellement les images avec des mots-clés, ce qui représente un travail long et difficile.

Dans ce TP, vous allez implémenter un petit système de recherche d'images par le contenu.. Le système que vous développerez exploitera plusieurs caractéristiques : d'histogrammes couleurs, textures (matrices de co-occurrences), formes (Moments de Hu)…

Le système calculera la distance entre plusieurs images d'une base et affichera la liste des images triées de la plus proche à la plus lointaine.

Aucune interface graphique n'est attendue pour ce TP et les résultats seront affichés sous la forme du nom de fichier le plus proche.

En entrée, votre programme devra prendre comme Argument (input) le nom d'une image requête. Alors, le système devra calculer une signature et une distance entre cette image et toutes les images de la base. La sortie de votre programme affichera les N images les plus proches (plus petites distances).

Exemple :

> search img035.tif

Exemple de sortie (avec N=5) :

Image	*Distance*
img078.tif	***0.001***
img032.tif	***0.005***
img005.tif	***0.010***
img098.tif	***0.012***
img062.tif	***0.016***

Les étapes de votre système sont :

- Lire l'image d'entrée

- Calculer les caractéristiques (couleur) + Moments de Hu
- Supprimer l'image requête de la mémoire (ne garder que sa description)
- Pour chaque image de la base :
  + Répéter les étapes ci-dessus

- Calculer une distance entre le vecteur de l'image requête et chacune des images de la base
- Afficher les N noms de fichiers correspondant aux plus petites distances (images les plus proches)

Pour éviter de surcharger la mémoire, il ne faut jamais avoir plus qu'une image en mémoire à la fois. Seul un tableau contenant les distances et les noms des fichiers est conservé en mémoire. On peut aussi travailler en 2 étapes où (1) toutes les caractéristiques sont pré-calculées et écrites dans un fichier et (2) les distances sont calculées à partir du fichier de caractéristiques.

Description de la base d'images

Vous allez utiliser la base COIL (Columbia University Image Library).
Elle contient 100 différent objets avec 20 images pour chaque objet sous différentes prises de vues.
Elle est librement téléchargeable ici :
http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php

Caractéristiques

Intersection d'histogrammes couleurs

La première caractéristique que nous allons utiliser est la couleur. Plusieurs méthodes existent pour comparer les couleurs, et nous retiendrons une méthode simple mais qui fonctionne bien, l'intersection d'histogrammes couleurs. Deux étapes sont nécessaires pour utiliser cette méthode

1. Calcul d'un histogramme

Pour simplifier les calculs, nous utiliserons l'espace RVB pour ce TP, mais sachez que d'autres espaces couleurs peuvent donner de meilleurs résultats de comparaison entre images.
Pour calculer un histogramme couleur, on calcule trois histogrammes, un sur chacun des canaux couleur. On obtient donc 3 histogrammes x 256 valeurs chacun. C'est beaucoup trop pour la comparaison des couleurs, donc nous réduirons cet histogramme à 3 couleurs x M valeurs chacun (3 x M valeurs au total). La réduction se fait simplement en effectuant une quantification plus forte de l'histogramme. Les valeurs de M utilisées sont couramment de 16, 24 ou 32. Dans ce TP, choisissez la valeur de votre choix. Cette valeur est un argument d'entrée de votre programme (-m).
Pour réduire un histogramme de 256 à M valeurs, on découpe l'histogramme en morceaux de 256/M valeurs, et on additionne les valeurs pour chaque morceau pour obtenir l'histogramme réduit (equal width discretization).

Pour aller plus vite, n'hésitez pas à utiliser les fonctions proposées par la bibliothèque

que vous utilisez

2. Intersection de 2 histogrammes

Pour chaque image, il y a un histogramme réduit de taille 3 x *M* qui est calculé. L'étape suivante est de faire l'intersection (distance) entre 2 histogrammes réduits pour deux images.

Pour cela, il est possible d'utiliser la formule suivante :

$$\frac{\sum_{i=0}^{3xM} |Histo1(i) - Histo2(i)|}{\sum_{j=0}^{3xM} Histo1(j)}$$

où *Histo1* représente l'histogramme réduit de l'image requête et *Histo2* l'histogramme réduit de l'image courante. Le résultat de cette formule devrait être entre 0 et 1 et représente la distance couleur entre les deux images.

Tout autre mesure est possible, n'hésitez pas à fouiller dans la documentation de la bibliothèque que vous utilisez !!!

Caractéristiques de texture : matrice de co-occurence

La deuxième caractéristique utilisée pour la comparaison entre les images est la texture. Pour chaque image, 4 matrices de co-occurrences seront calculées sur les images en niveaux de gris pour une distance=1 et pour quatre direction (0, 45, 90 et 135 degrés). Cinq étapes sont nécessaires ici pour obtenir une version simple mais fonctionnelle du calcul de la distance entre les textures :

1. Conversion de l'image en niveaux de gris

Nous calculerons les textures uniquement sur l'image en niveau de gris pour simplifier la tâche. Pour convertir les images en niveaux de gris il suffit d'utiliser (pour chaque pixel) la fonction suivante :

gris = (rouge + vert + bleu) / 3

2. Réduction des niveaux de gris de l'image

Comme pour les couleurs, il est inutile de conserver tous les niveaux de gris dans le calcul des textures. Nous réduirons la quantification de l'image pour la faire passer de 256 à T niveaux de gris (T=8, 16 ou 24). Cette valeur est un argument d'entrée de votre programme (-t). Pour réduire les niveaux de gris, il suffit de diviser chaque pixel de l'image par T et de conserver la partie entière du résultat.

3. · Calcul des matrices de co-occurrences

Rappel de la formule pour le calcul d'une matrice de co-occurrences:

$$C_{\Delta x,\Delta y}(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m} \begin{cases} 1, & \text{if } I(p,q) = i \text{ and } I(p+\Delta x, q+\Delta y) = j \\ 0, & \text{otherwise} \end{cases}$$

Ici, 4 matrices de co-occurrences par image doivent être calculées, pour une distance donnée et pour des directions=0, 45, 90 et 135 degrés. Normalement, une seule fonction est nécessaire pour calcul, si on passe en paramètres dx et dy, les distances x et y pour le calcul (exemple pour une distance 1 : pour direction=0 degré, dx=1 et dy=0 ; pour direction=45 degrés, dx=1 et dy=1 ; pour direction=90 degrés, dx=0 et dy=1 ; pour direction=135 degrés, dx=-1 et dy=1). Le nombre de matrices est à votre choix, mais je vous suggère d'en prendre minimum 8, c'est-à-dire quatre directions (0, 45, 90 et 135 degrés) et deux distances (1 et 2).
Attention ! Pensez à normaliser chaque matrice de co-occurence après calcul. Pour normaliser une matrice, on divise simplement par le nombre d'éléments (paires de pixels) de cette matrice. Vous aurez à calculer des matrices de tailles différentes, et elles ne peuvent être comparées que si elles sont bien normalisées.

4. · Calculs des paramètres sur les matrices de co-occurrences

Une fois les matrices de co-occurrence normalisées, vous pouvez calculer entre 4 et 14 mesures par matrice. Il est conseillé de prendre les 4 mesures suivantes pour commencer :

$$ENERGIE = \sum_{j}\sum_{i}(p(i,j))^2$$

$$INERTIE = \sum_{j}\sum_{i}(i-j)^2\, p(i,j)$$

$$ENTROPIE = -\sum_{j}\sum_{i} p(i,j)\log(p(i,j))$$

$$MOMENT\ DIFFERENTIEL\ INVERSE = \sum_{i}\sum_{j}\frac{1}{1+(i-j)^2}\, p(i,j)$$

5. · Calcul de la distance entre les textures de deux images différentes

Dernière étape, il faut calculer la distance entre les textures pour deux images différentes. Pour cela, nous utiliserons la formule simple suivante :

$$Dist(im1,im2) = \frac{\sqrt{\sum_{i=1}^{4}(param_i(im1) - param_i(im2))^2}}{4}$$

où $Param_i$ est une mesure (énergie, entropie, inertie ou moment différentiel inverse), et param(im1) est le paramètre calculé sur l'image requête et param(im2) le paramètre calculée sur l'image 2. Le résultat pour chaque mesure est une distance entre 0 et 1.

Calcul des moment de Hu (descripteur de forme)

La troisième vcaractéristique que vous allez utiliser pour comparer 2 images est un descripteur de forme : les moments de Hu. Pour chaque image, ces moments vont résumer la forme de l'objet. Ils sont calculés à partir des images en Niveaux de gris (utilisées précédement)

1. **Conversion de l'image en niveaux de gris**

Voir précédent

2. **Réduction des niveaux de gris de l'image**

Voir précédent

3. **Calcul des moments et des caractéristiques**

Vous devez dans un premier temps calculer les moments de l'image poru obtenir les moments centrés et normés

moments

$$m_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} I(x,y)x^p y^q dxdy \quad p \in \ ^+, q \in \ ^+ \tag{1}$$

Moments centrés

$$M_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} I(x,y)(x-x_0)^p (y-y_0)^q dxdy \quad p \in \ ^+, q \in \ ^+ \tag{2}$$

Moments normalisés

$$\mu_{pq} = \frac{M_{pq}}{M_{00}^{1+\frac{p+q}{2}}} \quad aussi\ noté\ \frac{M_{pq}}{M_{00}^{\gamma}}\ avec\ \gamma = \frac{p+q}{2}+1 \tag{3}$$

À partir des ces moments, vous pouvez calculer les 7 caractéristiques de Hu qui formeront votre vecteur de caractéristiques pour comparer les images.

$$(HMI)_1 = I_{1,1} = \mu_{02} + \mu_{20} \qquad (HMI)_1$$

$$(HMI)_2 = \left|I_{2,0}\right|^2 = \left(\mu_{20} - \mu_{02}\right)^2 + 4\mu_{11}^{\ 2} \qquad (HMI)_2$$

$$(HMI)_3 = \left|I_{3,0}\right|^2 = \left(\mu_{30} - 3\mu_{12}\right)^2 + \left(3\mu_{21} - \mu_{03}\right)^2 \qquad (HMI)_3$$

$$(HMI)_4 = \left|I_{2,1}\right|^2 = \left(\mu_{30} + \mu_{12}\right)^2 + \left(\mu_{21} + \mu_{03}\right)^2 \qquad (HMI)_4$$

$$\begin{aligned}(HMI)_5 &= 2\left|I_{3,0}\right|\left|I_{2,1}\right|^2 \cos\left(\theta_{3,0} - 3\theta_{2,1}\right) \\ &= \left(\mu_{30} - 3\mu_{12}\right)\left(\mu_{30} + \mu_{12}\right)\left[\left(\mu_{30} + \mu_{12}\right)^2 - 3\left(\mu_{21} + \mu_{03}\right)^2\right] \\ &+ \left(3\mu_{21} - \mu_{03}\right)\left(\mu_{21} + \mu_{03}\right)\left[3\left(\mu_{30} + \mu_{12}\right)^2 - \left(\mu_{21} + \mu_{03}\right)^2\right]\end{aligned}$$

$$(HMI)_6 = 2\left|I_{2,0}\right|\left|I_{2,1}\right|^2 \cos\left(\theta_{2,0} - 2\theta_{2,1}\right)$$
$$= \left(\mu_{20} - \mu_{02}\right)\left[\left(\mu_{30} + \mu_{12}\right)^2 - \left(\mu_{21} + \mu_{03}\right)^2\right] + 4\mu_{11}\left(\mu_{30} + \mu_{12}\right)\left(\mu_{21} + \mu_{03}\right)$$
$$(HMI)_7 = \left|I_{40}\right|^2 = \left(3\mu_{21} - \mu_{03}\right)\left(\mu_{30} + \mu_{12}\right)\left[\left(\mu_{30} + \mu_{12}\right)^2 - 3\left(\mu_{21} + \mu_{03}\right)^2\right]$$
$$-\left(\mu_{30} - 3\mu_{12}\right)\left(\mu_{12} + \mu_{03}\right)\left[3\left(\mu_{30} + \mu_{12}\right)^2 - \left(\mu_{21} + \mu_{03}\right)^2\right]$$

4. Calcul de la distance entre 2 images

Comme précédemment, il est possible une distance classique pour comparer 2 vecteurs :

$$Dist(im1, im2) = \frac{\sqrt{\sum_{i=1}^{7} (param_i(im1) - param_i(im2))^2}}{7}$$

Fonction gloabale de similarité entre 2 images

Nous avons maintenant 3 distances : couleur, texture et forme. La fonction globale de similarité entre deux images sera la somme pondérée de ces trois distances :

$Distance = \omega 1 * dist(couleur) + \omega 2 * dist(texture) + \omega 3 * dist(forme)$

avec $\omega 1 + \omega 2 + \omega 3 = 1$. (par défaut $\omega 1 = \omega 2 = \omega 3 = 0{,}33$)