Atelier Base de Données Multimédia

Le but de cet atelier est de découvrir c'est quoi un système de recherche dans une base de données multimédia, tester quelques descripteurs sur lesquels se base et évaluer ses performances.

Avant de commencer !

Créer votre compte Matlab en ligne, et préparer un workspace.

Créer un dossier « image_database » et coller les 66 images existent dans le répertoire local[1].

Partie 01 : Similarité avec le descripteur couleur :

But :

Le but de ce TP se résume comme suit :

_ Calculer la similarité entre deux images.

_ Chercher une image dans une base de données d'images grâce à une requête image lancée par l'utilisateur en se basant sur le descripteur couleur (Histogramme).

Algorithme [2]:

- ✓ Sélectionner une image requête à l'aide de la fonction uiget_le.
- ✓ Parcourir les 66 images en calculant les distances séparant l'image requête aux images de la base de données.
- ✓ Stocker les résultats des distances dans des vecteurs et trier-les par ordre croissant des distances.
- ✓ Afficher les 8 images les plus proches de l'image requête selon les trois canaux R, G et B.

---

[1] Dans les ressources délivrées avec l'atelier vous avez trouvé un dossier « Partie1 » qui contient le dossier des images à copier.

[2] On désigne par « Algorithme » une description générale de ce qui va passer dans chaque partie de l'atelier, donc ce n'est pas demander de faire ces étapes mais seulement pour prendre une vision globale avant d'entamer la section suivante « implémentation sur Matlab » dans laquelle il est mentionné le travail à faire.

Implémentation sur Matlab [3]:

_ Ajouter à votre « workspace » les scripts[4] suivants :

- ✓ « DescCouleur.m » : le programme principal (main).
- ✓ « distance » : la fonction qui calcule la distance Euclidienne.
- ✓ « distance_manhat » : la fonction qui calcule la distance Manhattan.
- ✓ « rgbhist » : la fonction qui calcule les histogrammes de l'image requête.

_ Commencer à tester avec plusieurs images requêtes, en utilisant une distance différente à chaque tentative.

_ En se basant sur les résultats qu'est-ce que vous remarquez !

Partie 02 : Similarité avec le descripteur texture :

But :

Le but de cet exercice est de rechercher des images dans une base de données contenant 66 images. Pour ce, on se focalise sur les descripteurs de texture statistiques comme un outil efficace à la recherche d'information.

Algorithme :

- ✓ Sélectionner une image requête.
- ✓ Convertit les images couleurs de la base de données en images de niveaux de gris et les stocke dans une matrice 3D.
- ✓ Écrire des fonctions calculant les descripteurs suivants :

[3] Dans cette section vous trouverez les étapes à suivre pour implémenter les scripts nécessaires pour chaque partie de l'atelier ainsi que le travail demandé à faire.

[4] Les scripts nécessaires pour chaque partie sont dans le dossier « partie{i} » avec i={1,2,3} , vous pouvez les consulter, ils sont commentés.

$$Variance = \frac{1}{l \times c} \sum_i \sum_j (M(i,j) - \overline{m})^2$$

$$Energy = \sum_i \sum_j M^2(i,j)$$

$$Entropy = -\sum_i \sum_j M(i,j) \log_2 M(i,j)$$

$$Contrast = \sum_i \sum_j (i-j)^2 M(i,j)$$

$$Homogeneity = \sum_i \sum_j \frac{M(i,j)}{1+|i-j|}$$

- ✓ Diviser chaque image en N régions distinctes, ensuite nous allons appliquer les descripteurs précédemment calculés sur les N régions. Le descripteur de chaque image est la concaténation de ces N valeurs calculées.
  Garder les descripteurs calculés dans la matrice de descripteurs selon les N régions.
- ✓ En utilisant ces descripteurs et en évaluant la similarité avec la distance Euclidienne, rechercher les 4 images les plus proches à l'image requête dans la base de données.

Implémentation sur Matlab :

_ Ajouter à votre « workspace » les scripts suivants :

- ✓ « DescTexture.m » : le programme principal (main).
- ✓ « stocker_images.m » : la fonction qui convertit les images couleurs de la base de données en images de niveaux de gris et les stocke dans la matrice 3D mat_image.
- ✓ « stocker_desc.m » : la fonction qui calcule la matrice de descripteurs des images sur les N régions..
- ✓ « distance_eucli.m » : la fonction qui calcule la similarité entre les descripteurs.
- ✓ « variance.m » « energie.m » « entropie.m » « contraste.m » « moment.m » : des scripts pour calculer les différents descripteurs statistiques.

_ Exécuter le script « DescTexture.m » et tester avec plusieurs images requêtes et différents descripteurs statistiques, qu'est-ce que vous remarquez !

Partie 03 : Evaluation des descripteurs :

But :

Le but de cet exercice est d'évaluer les performances des descripteurs présentés dans les chapitres précédents dans une nouvelle base de données contenant 60 images.
On va tester l'utilisation du descripteur couleur dans un premier temps et on va évaluer sa performance via le principe CMC, ensuite on va introduire les descripteurs statistiques qui seront présenter par une matrice co_occurence et on va les combiner avec le descripteur couleur puis on va tester la performance à nouveau.

Algorithme :

Soit une base de données de 60 images en niveaux de gris contenant les visages de 30 personnes différentes. Chaque personne est représentée par 2 visages ; un visage requête et un autre stocké dans la base de données. Le but est d'afficher la courbe CMC de plusieurs descripteurs en comparant leurs performances. Pour cela, il faut suivre les étapes suivantes :

- ✓ Stocker les 60 images dans une matrice 3D.
- ✓ Créer une matrice 2D contenant les descripteurs (histogrammes) des 60 images.
- ✓ Créer la matrice de distance contenant les distances séparant les 30 images requêtes des 30 images de la base de données.
- ✓ Calculer les CMC pour les descripteurs suivants :
  (a) Histogramme
  (b) Matrice de co-occurence
- ✓ Dessiner la courbe CMC comparant les performances de ces descripteurs.

Implémentation sur Matlab :

_ Créer un dossier « im » et importer les images[5].

_ Ajouter à votre « workspace » les scripts nécessaires :

- ✓ « EvaluerCMC.m » : le programme principal (main).
- ✓ « stocker_images_cmc.m » : la fonction qui stocke les images dans la matrice 3D mat_image.

[5] Les images sont dans le sous-dossier « im » du dossier « Partie 3 ».

- ✓ « stocker_desc_cmc.m » : la fonction qui calcule la matrice de descripteurs des images en se basant sur l'histogramme.
- ✓ « stocker_dist_cmc.m » : la fonction qui calcule la matrice des distances entre chaque image requête et les autres images de la BD.

- ✓ « distance_eucl.m » : la fonction qui calcule la similarité en se basant sur la distance.
- ✓ « calcul_CMC.m » : la fonction qui construit le tableau des rangs et tracer la courbe CMC.
- ✓ « co_occurence.m » : une fonction pour construire la matrice co_occurence[6].
- ✓ « histoN.m » : une fonction pour calculer l'histogramme normalisé.

_ Exécuter le script « EvaluerCMC.m » et choisir le descripteur couleur (histogramme) comme premier traitement, analyser la courbe CMC obtenue.

_ Choisissez le matrice co-occurrence comme deuxième traitement[7], Analyser la courbe CMC obtenue.

_ Comparer les deux courbes CMC. Qu'est-ce que vous remarquez concernant leurs performances[8] !

[6] C'est une matrice associée à chaque image, elle combine les descripteurs statistiques de cette image.
[7] Dans ce traitement les descripteurs statistique et couleur sont combinés.
[8] Celui qui converge rapidement vers la probabilité 1 est plus performant « pourquoi ? Voir la présentation ».