

TP3 : Indexation du contenu pictural

Présenté à

Hughes Perreault

Par Élisa Correia-Martins - 1880313

Dans le cadre du cours

INF8770 : Technologies multimédias

Département de génie informatique

Polytechnique Montréal

7 décembre 2020

**Provenance du code**

**Question 1**

*Alice, Bob et Carol sont trois étudiants n’ayant pas suivi le cours INF8770. Ils proposent cependant trois algorithmes permettant de retrouver une vidéo à partir d’une image.*

*Alice propose de stocker les vidéos sous le format MP4 telles quelles sans traitement particulier. Puis, étant donnée une image, elle propose de parcourir toutes les vidéos, trame par trame, et de calculer une distance euclidienne entre les pixels de l’image d’intérêt et la trame de la vidéo. Dès que la distance est nulle, elle retourne la vidéo en question ainsi que le minutage. Si la distance nulle n’est jamais atteinte, elle renvoie un message indiquant que la vidéo n’a pas été trouvée.*

*Bob propose de stocker les vidéos sous la forme d’images (et d’un fichier d’indexation optionnel) car il pense que cela accélère le traitement compte tenu qu’il s’agit de comparer des images avec d’autres images. Pour cela, il hésite entre les deux formats PNG et JPEG. Pour la suite, il reprend les idées d’Alice : il propose de parcourir toutes les trames extraites et de calculer une distance euclidienne avec l’image d’intérêt. Dès que la distance est nulle, il retourne la vidéo associée à la trame ainsi que le minutage. Si la distance nulle n’est jamais atteinte, il renvoie un message indiquant que la vidéo n’a pas été trouvée.*

*Carol propose de sélectionner une trame à chaque seconde de chaque vidéo et de la décrire par un histogramme de couleurs (un fichier d’indexation additionnel peut être utilisé). Puis, étant donnée une image, Carol calcule son histogramme de couleurs et retourne la vidéo et le minutage de l’histogramme de couleurs ayant la plus faible distance euclidienne.*

*D’après vos connaissances en technologies multimédias, discutez de la pertinence de chaque algorithme lors de l’application sur les images en PNG et en JPEG. Quelles recommandations feriez-vous à Bob quant au format de l’image dans son algorithme ?*

La méthode d’Alice n’est pas appropriée pour retrouver une vidéo à partir d’une image, car calculer la distance euclidienne pour chaque trame de chaque vidéo jusqu’à trouver la bonne serait beaucoup trop long et inefficace.

Le format PNG est un format d’image sans perte et le format JPEG est un format d’image avec perte. Pour que l’algorithme fonctionne correctement, il faut donc que Bob utilise le format PNG, puisque le format JPEG ne permettra pas d’obtenir une distance euclidienne nulle.

La méthode de Carol n’est pas non plus très adéquate. Tout d’abord, le fait de ne prendre qu’une trame par seconde. Il est possible qu’une image soit extraite d’une scène qui dure moins d’une seconde et alors la scène ne serait pas représentée par les trames sélectionnées. De plus, les histogrammes de couleur ne permettent pas de décrire la sémantique de l’image, il serait alors possible que […]. Finalement, il se peut que l’image n’appartienne à aucun vidéo, la méthode de Carol retournera tout de même la vidéo la plus proche.

**Question 2**

*Implémentez l’un des trois algorithmes proposés et appliquez-le sur les images en PNG et en JPEG. Est-ce que les résultats en termes de performance sont conformes à vos remarques pour la question 1 ? NB1 : si vous choisissez l’algorithme de Bob, prenez le format d’images qui vous semble le plus pertinent. NB2 : si vous choisissez l’algorithme de Carol, vous être libres quant aux choix des paramètres de l’histogramme des couleurs.*

L’algorithme implémenté est celui de Carol et est situé dans le fichier carol.py. L’algorithme est conforme aux attentes. Tout d’abord, puisque Carol retourne la vidéo ayant la plus faible distance euclidienne, les images qui ne correspondent à aucune vidéo seront associée à une vidéo et alors on a automatiquement plusieurs mauvaises réponses.

**Question 3**

*Implémentez votre propre algorithme de reconnaissance de vidéos. Discutez de vos choix d’implémentation (sélection d’images clefs, format des données sauvegardées, algorithme de recherche, caractérisation d’une image, etc.) Comparez vos mesures de performances avec la méthode implémentée à la question 2 vis-à-vis des quatre mesures de performances sur les images en PNG et en JPEG. Synthétisez vos résultats dans un tableau ou un graphique.*

**Sélection d’images-clés**

La sélection d’images-clés se fait à partir de la méthode par groupement et comparaison d’histogrammes d’orientation des gradients. Tout d’abord, les filtres de Sobel sont appliqués sur chaque trame du vidéo pour obtenir les gradients, puis on calcule leur histogramme des orientations du gradient selon la fréquence de chaque angle quantifié multiplié par la force du gradient. On groupe ensuite les histogrammes selon une distance euclidienne. La méthode de groupement implémentée suit la méthode de la diapositive 27 du chapitre 8 vu en cours. Tout d’abord, on crée un premier groupe avec l’histogramme de la première trame comme centre de masse. Ensuite, on calcule pour chaque trame la distance euclidienne avec le centre de masse de chaque groupe. On ajoute la trame au groupe dont la distance euclidienne est la plus basse et en-dessous d’un seuil. Si la distance est au-dessus de ce seuil, on créé un nouveau groupe. À chaque itération, le centre de masse du groupe où on a ajouté la trame est mise à jour. Après le groupement, on sélectionne la trame avec l’histogramme le plus proche du centre de masse. Avec un seuil élevé, on obtiendra plus d’images-clés puisqu’il y aura plus de groupes qui seront créés et alors on a une meilleure probabilité de trouver la bonne réponse. Cependant, cela augmentera significativement le temps de résolution.

**Format des données sauvegardées**

Les informations sur images-clés sont enregistrées dans un fichier CSV, chaque ligne correspond à une image-clé et contient le numéro du vidéo auquel elle est associée, son minutage et son histogramme de gradient.

**Algorithme de recherche**

L’algorithme de recherche prend en entrée une image. Il calcule ensuite son histogramme d’orientation de gradient avec la fonction utilisée pour calculer les histogrammes des images-clés. Ensuite, on calcule itérativement la distance euclidienne entre les histogrammes de l’image en question et chaque image-clé. La fonction retourne la vidéo correspondant à l’image-clé dont la distance euclidienne de l’image à trouver est la plus petite. Puisque certaines image à trouver n’ont pas de vidéo correspondant, on vérifie en plus que la distance la plus petite est en-dessous d’un certain seuil. Ce seuil diffère selon le format de l’image. Ainsi, pour une image PNG, le seuil est défini à 7000 puisqu’il s’agit d’un format d’image sans perte, on tolère moins les distances euclidiennes élevées.

Caractérisation d’une image

Mesures de performances :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Algorithme de Carol | | Algorithme question3 | |
| PNG | JPEG | PNG | JPEG |
| Temps d’exécution moyen pour une image (en seconde) | 1.472 | 1.455 | 2.77 |  |
| Pourcentage de vidéos correctement reconnues | 77.5 | 73.0 | 94.0 |  |
| Écart (en valeur absolue) moyen sur le minutage lorsque la vidéo est correctement reconnue et présente dans la banque de données (en seconde) ; | 1.551 | 3.022 | 2.037 |  |
| Mémoire utilisée pour stocker sur le disque les informations sur les vidéos (en Mo) |  |  |  |  |

DC\_50\_97