SIM-SysIn : Test Final, partie machine

(Durée 1h20, documents autorisés : fichiers *.pdf et *.py vus en cours et TD, notes manuscrites)

Nom	*Prénom*	*Groupe TD*	*Date*
			15 décembre 2011

Le travail sur machine est à faire en utilisant la **clef USB** remise par l'enseignant, que vous devrez rendre en fin de session. Vous devez réaliser, dans l'ordre, les actions suivantes :

1 Renommage

Renommer le répertoire `TestFinalSysIn` de la clef USB en utilisant votre `Nom.prenom` comme nouveau nom de répertoire (pas d'accent, pas d'espace!).

Le répertoire renommé `Nom.prenom` contient une application graphique de traitement d'image, constituée d'un programme principal (fichier `main.py`) et de modules Python contenant les classes utiles (fichiers `ModMainWindow.py`, `ModImageArray.py` et `ModFrameImage.py`) .

L'outil à utiliser pour éditer et exécuter les fichiers Python est IDLE (vu en TD).

2 Codage du traitement `Contour`

On se propose d'implémenter au sein de l'application graphique un traitement d'extraction de contour. Ce type de traitement se décompose en le calcul, pour tout pixel i de l'image, du gradient horizontal, Gh_i, et du gradient vertical, Gv_i. La valeur du pixel i est alors calculée par : $p_i = \sqrt{Gh_i^2 + Gv_i^2}$ (éq. 1)

Pour implémenter ce traitement, vous devez scrupuleusement suivre les étapes suivantes :

2.1 Codage dans la Classe MainWindow

La première étape consiste à ajouter les sous-menus `'GradientH'`, `'GradientV'` et `'Contour'` au menu `'Transformer'`, et à les connecter respectivement aux méthodes privées `HGradient`, `VGradient` et `Edge`.

sous-menus :

À la fin du constructeur de la classe `MainWindow`, copier/coller trois fois la ligne `menuT.addAction('Monochrome', self.__UniColor)` et la modifier pour créer les sous-menus `'GradientH'`, `'GradientV'` et `'Contour'` associés respectivement aux méthodes privées `HGradient`, `VGradient` et `Edge` (contour est traduit en anglais par *edge*) .

méthodes privées :

Créer les méthodes privées `HGradient`, `VGradient` et `Edge` par copier/coller de la méthode privée `Negative` .

Par quel mécanisme les actions de ces 3 sous-menus conduisent-elles à exécuter des méthodes publiques *adhoc* de l'attribut privé `ARGBarray` ?

Tests préliminaires :

Sous **IDLE**, vous pouvez lancer l'application, et vérifier la présence des 3 sous-menus. Si vous essayer de lancer un des 3 nouveaux sous-menus vous obtenez un message d'erreur, expliquez le problème :

2.2 Codage dans la Classe ImageArray

Il faut implémenter dans cette classe les trois méthodes publiques `HGradient`, `VGradient` et `Edge`. Les algorithmes sont les suivants (calculs des gradients par différences finies centrées d'ordre 1) :

Méthode HGradient

1. Copier les valeurs 3 tableaux privés `self.__R`, `self.__G` et `self.__B` dans 3 tableaux locaux `R`, `G` et `B` (méthode `copy` de la classe `ndarray`),
2. boucle sur les lignes des tableaux, de la deuxième à l'avant dernière
3. boucle sur les colonnes des tableaux, de la deuxième à l'avant dernière
4. calculer chaque élément `X[r, c]` (`X` représentant successivement `R`, `G` et `B`) selon :
   `X[r, c]= 0.5*(Y[r, c+1]-Y[r, c-1])`
   (`Y` représentant successivement `self.__R`, `self.__G`, et `self.__B`)
5. fin des boucles
6. affecter les 3 tableaux locaux `R`, `G` et `B` aux 3 tableaux privés `self.__R`, `self.__G` et `self.__B` (simple égalité !).

Pourquoi est-ce important de faire une copie des tableaux et d'affecter le calcul de `X[r, c]` à la copie ?

Coder et tester.

Méthode VGradient

Exactement le même principe que précédemment, sauf la formule de calcul de l'élément `X[r, c]` :

```
X[r, c]= 0.5*(Y[r+1, c)-Y[r-1, c])
```

Coder et tester.

Méthode Edge

1. sauver l'état initial des 3 tableaux privés `self.__R`, `self.__G` et `self.__B` en les copiant dans 3 tableaux locaux `R`, `G` et `B` (méthode `copy` de `ndarray`)
2. exécuter la méthode `HGradient`
3. sauver les résultats du calcul précédent en copiant les 3 tableaux privés `self.__R`, `self.__G` et `self.__B` dans 3 tableaux locaux `Rh`, `Gh` et `Bh` (`copy` de `ndarray`)
4. remettre dans les 3 tableaux `self.__R`, `self.__G` et `self.__B` les valeurs copiées en 1. dans les 3 tableaux locaux `R`, `G` et `B` (`copy` de `ndarray`)
5. exécuter la méthode `VGradient`
6. sauver les résultats du calcul précédent en copiant les 3 tableaux privés `self.__R`, `self.__G` et `self.__B` dans 3 tableaux locaux `Rv`, `Gv` et `Bv` (`copy` de `ndarray`)
7. Calculer le résultat final :
   boucle sur les lignes des tableaux, de la deuxième à l'avant dernière
8. boucle sur les colonnes des tableaux, de la deuxième à l'avant dernière
9. calculer chaque élément `X[r, c]` (`X` représentant successivement `R`, `G` et `B`) selon l'équation 1 énoncée plus haut, au début du paragraphe 2.
10. fin des boucles

2.3 Tests

À faire : charger l'image `Smiley.png`, la transformer avec `Contour` puis `Negatif`, et sauver l'image obtenue dans un fichier nommé `SmileyContour.png`. Ce fichier servira à `l'évaluation` de l'implémentation du traitement `Contour`.

3 Codage du traitement `Histogram`

Ce traitement vise à tracer les histogrammes des 3 valeurs rouges, verts et bleus associées aux pixels d'une image transformée.

La méthode proposée est de faire tracer l'histogramme dans un fichier temporaire qui sera chargé dans l'objet privé `__cadre3` de type `ImageFrame`, contenu dans la classe `MainWindow`. Une fois chargé, le fichier pourra être effacé.

3.1 Classe MainWindow

sous menu :

À la fin du constructeur de la classe `MainWindow`, copier/coller, puis modifier la ligne `menuT.addAction('Contour', self.__Edge)` pour ajouter le sous-menu `'Histogramme'` au menu `'Analyser'`, et lui associer la méthode privée `Histogram`.

méthode privée `Histogram` :

Il faut maintenant implémenter la méthode privée `Histogram`, qui effectue les opérations suivantes :

1. affecter le nom du fichier histogramme à la variable locale `fileName` (par exemple `fileName='h.png'`)
2. exécuter la méthode publique `Histogram` de l'objet `__ARGBarray` , en lui passant les 2 arguments (`fileName`, et `20` comme nombre de classes).
3. créer un objet `QImage` à partir du fichier histogramme, et le faire afficher par la méthode `DisplayImage` de l'objet privé `cadre3` :
   ```
   image = QImage(fileName)
   self.__cadre3.DisplayImage(image)
   ```
4. effacer le fichier histogramme (fonction `remove()` du module `os` ; à ne faire qu'à la fin quand tout marche...)

3.2 Classe ImageArray

Le tracé sera fait par la méthode `Histogram`, qui prend 2 arguments :

1. le nom de fichier dans lequel sera tracé l'histogramme (argument `fileName`, valeur par défaut : `"hist.png"`)
2. le nombre de classes des histogrammes (argument `nbClasse`, valeur par défaut : `10`)

Le séquencement des opérations effectuées par la méthode `Histogram()` est le suivant :

1. import du module `matplotlib.pyplot` sous le nom `plt;` import de la classe `array` du module `numpy`
2. appel à la fonction `figure` du module `plt`, pour créer une nouvelle figure, avec comme argument `'figsize=(5,3.7)'`
3. préparation de la liste des tableaux :
   `tabLoc = array([self.__B.flatten(),self.__G.flatten(),self.__R.flatten()]).transpose()`
4. appel à la fonction `hist()` du module `plt`, avec les arguments :
   - `tabLoc`, la liste des tableaux analyser
   - le nombre de classes : `20`,
   - le type d'histogramme : `histtype='bar'`,
   - la plage des valeurs : `range=(0.,1.)`,
   - l'option de normalisation : `normed=True`.
5. sauvegarde du tracé par appel à la fonction `savefig` du module `plt`, avec les arguments :
   nom du fichier, suivi des arguments : `format='png', transparent=True, dpi=80, bbox_inches='tight'`

3.3 Tests

Sous `IDLE`, vous pouvez maintenant lancer l'application, charger une image du répertoire `Nom.prenom`, essayer des transformations et tracer les histogrammes associés.

À faire : charger l'image `two.png`, la transformer avec `Monochrome` pour obtenir des images rouge, verte et bleue ; tracer les histogrammes à chaque fois, et sauver sous les noms `twoHistoR.png`, `twoHistoG.png` et `twoHistoB.png`. Ces fichiers serviront à `l'évaluation` de l'implémentation du traitement `Histogramme`.