

MASTER M2TI

TP DE SEGMENTATION D'IMAGES

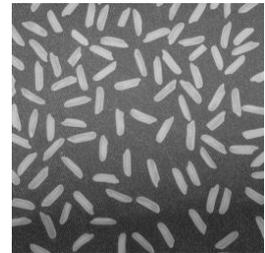
2016-2017



## TP4 – Exercices avancés de Morphologie

### Exercice 1 : Caractéristiques des grains de riz

Nous continuons de travailler sur l'image *rice.png*, et plus particulièrement sur l'image seuillée et nettoyée obtenue lors de la dernière séance.

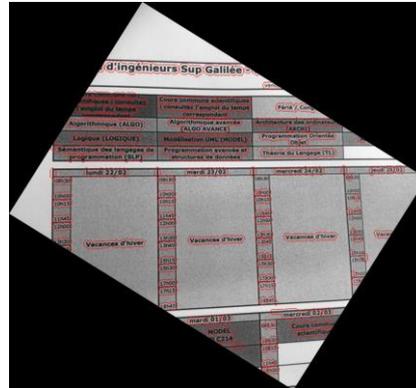
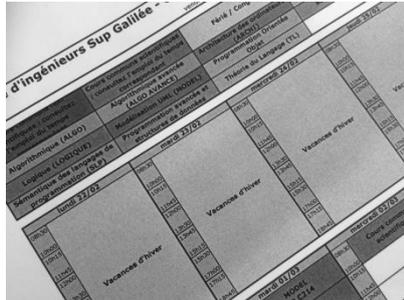


1. Commencez par supprimer tous les grains de riz touchant le bord de l'image.
2. Comptez et affichez le nombre de grains de riz restant sur l'image.
3. Pour chaque grain de riz, affichez sa taille (en pixels), la largeur approximative et sa longueur.
4. Générez une image couleur où apparaissent en bleu les grains de riz les plus courts, et en rouge ceux les plus longs.

Indice : `cv2.COLORMAP_JET` est une carte de couleur (colormap) qui devrait correspondre à ce que l'on souhaite obtenir. Voir <http://docs.opencv.org/2.4/modules/contrib/doc/facerec/colormaps.html>

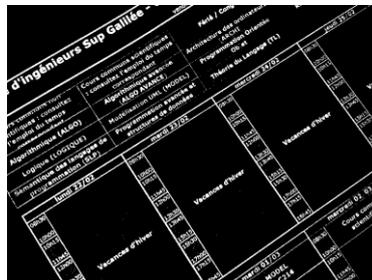
## Exercice 2 : Emplois du temps

Ouvrez et regardez l'image *feuille1.png* : le but sera d'extraire les éléments textuels de l'emploi du temps, ainsi que de le redresser automatiquement.



A gauche, l'image de départ. A droite, l'objectif que l'on se fixe.

1. Le texte qui nous intéresse est foncé sur fond un peu plus clair, mais qui n'est pas du tout uniforme. Quelle transformation devez-vous effectuer pour mettre en valeur le texte et uniformiser le fond ?
2. Cherchez un seuil satisfaisant permettant de séparer les pixels en deux groupes : texte et lignes dans un groupe, fond dans l'autre groupe. On travaille désormais avec une image binaire.



Le résultat que vous devriez avoir

3. Réalisez des ouvertures avec une ligne de 40 pixels que vous ferez tourner entre -90 et 90 degrés. L'ouverture permettant de conserver le plus grand nombre de pixels vous permettra de déduire l'orientation de l'image que vous conserverez pour effectuer un redressement automatique plus tard.
4. Grâce à une ouverture d'une ligne de 10 pixels (et d'orientation égale à l'angle trouvé précédemment), effacez les lignes de l'image.



Le résultat que vous devriez avoir

5. Si nécessaire, « nettoyez » le masque binaire en retirant les pixels de petite taille, puis procédez à une dilatation du masque binaire afin d'obtenir un effet d'entourage autour du texte, comme montré sur l'image ci-dessous.
6. Pour finir, dans l'image couleur originale, coloriez tous les pixels d'entourage en rouge, et effectuez une rotation de l'image afin de la redresser (voir le fichier myimage.py).
7. Essayez votre algorithme sur les autres images.