

IMI – Analyse d'image – S8

TP – Transformée de Hough

Marion Foare et Maxime Di Folco

Objectifs.

- Implémentation de la transformée de Hough pour la détection de droites vue en cours.
- Utilisation de la transformée de Hough généralisée pour la détection de caractères brailles sur un document scanné.

Déroulement. Ce TP se déroule sur **1 séance** et est à effectuer par binôme sous Matlab/Octave. Vous trouverez sur CPe-campus une archive contenant l'ensemble des fichiers nécessaires à la réalisation de ce TP.

Configuration. Ce TP nécessite les librairies suivantes :

- Matlab : *Image Processing Toolbox*
- Octave : *Image toolbox* pkg install -forge image
puis, en début de script : pkg load image

Evaluation. Dans un délai d'une semaine après la séance de TP, vous déposerez sur le dépôt dédié **une archive** nom1_nom2_hough.zip contenant tout le nécessaire pour reproduire vos résultats, à savoir :

- l'ensemble des images nécessaires à l'exécution du code;
- un script Matlab/Octave (au format .m) permettant l'exécution de l'ensemble du TP. On veillera à ce que ce script soit **correctement commenté** et que chaque section du TP soit **clairement identifiée** via l'utilisation de %%;
- un compte-rendu de **4 pages maximum au format pdf**, dans lequel vous commenterez :
 - ▷ votre démarche (algorithme, équations sur lesquelles vous vous êtes appuyés, etc.),
 - ▷ le choix de vos paramètres,
 - ▷ les résultats obtenus,
 - ▷ l'influence de la méthode d'extraction des contours sur les résultats,

- ▷ l'influence de la méthode d'extraction des maxima locaux sur les résultats,
- ▷ ainsi qu'une conclusion sur la méthode implémentée

Ce TP a pour vocation d'être formateur, et peut nécessiter de travailler en groupe (fortement encouragé, qui plus est!). Toutefois, il vous est demandé **un compte-rendu par binôme**, correspondant à **votre** restitution des notions. Il en va de même pour le code. Tout travail emprunté à un (ou plusieurs) autre(s) groupe(s) doit être **explicitement identifié**.

Nous rappelons que toute tentative de copie entraînera une sanction de l'ensemble des binômes concernés.

1 Détection de droites

Cette section consiste en l'implémentation de l'algorithme de la transformée de Hough vu en cours, pour la détection des droites dans l'image 'buildings.png'. Le code devra être structuré de la manière suivante :

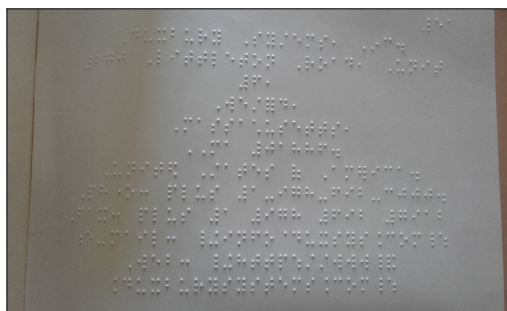
1. lecture de l'image
2. définition des paramètres de l'algorithme
3. création de l'image des contours
4. calcul de la matrice d'accumulation
5. extraction des maxima locaux
6. calcul des points extrémaux de chaque droite (par souci de simplification, on choisira des points du bord de l'image)
7. affichage des droites détectées sur l'image d'origine

Pour aller plus loin... vous pouvez tester votre code sur une autre image (proposée ou non), et/ou implémenter l'une des améliorations vues en cours.

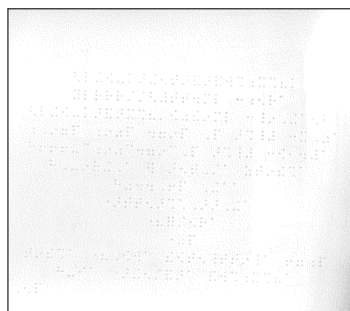
2 Détection de cercles

L'alphabet braille est un alphabet en relief (aussi dit tactile) utilisé par les personnes mal-voyantes et non-voyantes. Chaque caractère est composé d'une matrice de 6 points de taille identique, répartis sur 2 colonnes, et s'identifie par le nombre et la position des points en relief.

Une problématique actuelle consiste à lire automatiquement (par un ordinateur par exemple) un texte issu d'une version papier. Le figure 1 montre deux exemples de textes scannés, où l'on peut clairement distinguer les caractères en relief, malgré leurs aspects différents.



braille1.png



braille2.png

FIGURE 1 – Exemples d'images de textes en braille (avec l'aimable autorisation de B. Mascaret).

En utilisant les fonctions `imfindcircles` et `viscircles`, proposer une solution de détection des points en relief des caractères en braille dans les images 'braille1.png' et 'braille2.png', qui constituerait un pré-traitement satisfaisant pour une future détection des caractères eux-mêmes. La solution proposée pourra, si nécessaire, faire intervenir des outils de traitement d'image extérieurs à ce cours pour retravailler les images en amont de la détection de cercles.