

TS225

Compte rendu - Projet en Traitement des Images

Maxime PETERLIN - `maxime.peterlin@enseirb-matmeca.fr`

Gabriel VERMEULEN - `gabriel@vermeulen.email`

ENSEIRB-MATMECA, Bordeaux

1^{er} janvier 2014

Contents

1	Détermination de la région d'intérêt	2
2	Estimation de la signature	3
3	Identification des chiffres	4
4	Cas des codes-barres <i>orientés</i>	4
5	Traitement de la base de données des codes-barres	4

1 Détermination de la région d'intérêt

La première partie de ce projet a été la détection quasi-automatique de la région d'intérêt du code-barres, i.e. le code-barres sans les zones blanches l'entourant, ni les différents chiffres se trouvant en dessous de ce dernier.

Nous avons restreint nos premiers tests au code-barres suivant.



Figure 1: code-barres

On notera que ce code-barres est droit et que l'image le représentant est nette.

Afin de simplifier le traitement et la lecture du code-barres, on passe l'image en niveau de gris en moyennant les composantes R, G et B de chaque pixel.

Le principe de détection repose sur deux actions au préalable de l'utilisateur. En effet, il devra déterminer manuellement les bords droits et gauche du code-barres grâce à la fonction **ginput** de Matlab. Le programme se chargera ensuite de déterminer l'extension horizontale du code-barres.

On appelle $A(x_1, y_1)$ et $B(x_2, y_2)$ les deux points déterminés par l'utilisateur. Grâce à ces derniers, on peut déterminer la région \mathcal{R}_0 dont les coins supérieur gauche et inférieur droit ont pour coordonnées respectives (x_{min}, y_{min}) et (x_{max}, y_{max}) . L'objectif, à présent, est d'étendre cette région verticalement, pour arriver à la région d'intérêt \mathcal{R}_T .

Pour ce faire, on part de la ligne de numéro y_{min} et on somme toutes les composantes de cette dernière ayant des abscisses comprises entre x_{min} et x_{max} . On va ensuite décrémenter la valeur du numéro de ligne y qui était initialement à y_{min} et on réitère l'opération précédente sur la ligne actuelle. On compare alors les résultats obtenus et suivant la différence entre les lignes on considère ou non que l'on est sorti de la zone d'intérêt. On obtient finalement notre nouveau y_{min} .

Le même raisonnement peut s'appliquer pour la recherche de la nouvelle valeur de y_{max} .

Cette méthode est résumée par la formule mathématique suivante, où I est la matrice représentant l'image du code-barres et ϵ un paramètre de tolérance.

$$1 - \epsilon < \left| \frac{\sum_{x=x_{min}}^{x_{max}} I(x, y)}{\sum_{x=x_{min}}^{x_{max}} I(x, y_{min})} \right| < 1 + \epsilon$$

Dans le cadre de notre exemple, on obtient le résultat suivant.



Figure 2: Région d'intérêt \mathcal{R}_T

2 Estimation de la signature

Dans cette partie, nous allons estimer la signature du code-barres qui est une représentation monodimensionnelle des informations portées par ce dernier.

En effet, en théorie, il suffit d'avoir une seule ligne du code-barres pour pouvoir le décoder. Cependant, nous allons nous servir de cette redondance d'information pour pallier aux différentes erreurs que le code-barres pourrait comporter (rayures, flou, etc...).

Nous utiliserons toujours le code-barres utilisé précédemment comme exemple dans cette partie.

On commence par projeter l'image sur l'axe horizontale. Cela se traduit par un moyennage des valeurs de chaque colonnes comprises entre x_{min} et x_{max} .

On obtient alors la signature suivante.



Figure 3: Signature non seuillée du code-barres

Ensuite, on va binariser l'image pour qu'elle ne soit formée plus que par des pixels blancs ou noirs. Nous allons devoir trouver un seuil délimitant les valeurs qui nous intéressent et, pour ce faire, nous appliquerons l'algorithme de Otsu qui repose sur le calcul d'un histogramme h sur N classes (qui sont ici les 256 niveaux de gris).

On commence par calculer $w : k \mapsto w(k)$ et $\mu : k \mapsto \mu(k)$ avec $k \in \llbracket 0, N - 1 \rrbracket$.

Notre seuil s est alors donné par $s = \max_{k \in \llbracket 0, N - 1 \rrbracket} w(k)[\mu(N - 1) - \mu(k)]^2 + (1 - w(k))\mu(k)^2$

On peut, à présent, passer à l'étape de décision pour binariser l'image : 0 correspond à un pixel noir et 1 à un pixel blanc.

La prochaine étape est la suppression des bits inutiles sur les bords de l'image venant de la région d'intérêt sélectionnée approximativement par l'utilisateur. Maintenant que l'image a été binarisée, cette étape est simple, car il suffit de supprimer les pixels blancs les plus au bord de la signature. On obtient alors le résultat qui suit.

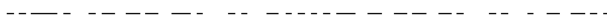


Figure 4: Signature seuillée du code-barres

3 Identification des chiffres

4 Cas des codes-barres *orientés*

5 Traitement de la base de données des codes-barres