

# TS225

## Compte rendu - Projet en Traitement des Images

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

Gabriel VERMEULEN - `gabriel@vermeulen.email`

ENSEIRB-MATMECA, Bordeaux

1<sup>er</sup> 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-barress, i.e. le code-barress 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-barress suivant.



Figure 1: code-barress

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

Afin de simplifier le traitement et la lecture du code-barress, 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-barress grâce à la fonction **ginput** de Matlab. Le programme se chargera ensuite de déterminer l'extension horizontale du code-barress.

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-barress et  $\epsilon$  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-barress qui est une représentation monodimensionnelle des informations portées par ce dernier.

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

Nous utiliserons toujours le code-barress 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-barress

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

Maintenant que notre image est uniquement composée de pixels noirs et blancs, l'objectif de cette partie va être d'identifier les chiffres représentés par le code-barres.

Nous savons qu'un code-barres est composé de 95 éléments de taille unitaire (i.e. les lignes le représentant). La première étape va être la séparation de la ligne obtenue dans la partie précédente en ces 95 éléments. De plus, il va falloir identifier la couleur de chaque élément, car ces derniers sont composés de plusieurs pixels et il se peut que leur couleur ne soit pas uniforme, d'où la nécessité de prendre une décision sur cette dernière.

Notre approche est relativement directe dans le sens où la couleur d'un élément donné sera celle des pixels majoritaires de cette couleur et on peut alors ramener un ensemble de pixels à un seul bit pour décrire une barre.

On arrive alors à ramener la signature du code-barres obtenue à l'issue de la partie précédente en un vecteur de 95 bits.

L'étape suivante est la reconnaissance des chiffres. La signature d'un chiffre est donné par la concaténation de 7 éléments unitaires du code-barres. Dans le vecteur obtenu précédemment, les chiffres sont compris entre les indices 4 et 45, ainsi qu'entre les indices 51 et 92.

Pour identifier les chiffres, nous utiliserons une matrice de 30 lignes et 7 colonnes comportant les signatures théoriques des chiffres. Nous comparons alors la signature observée à la matrice des signatures théoriques pour obtenir la valeur du chiffre encodée.

La critère utilisé est le suivant.

$$c(s_{th}, s_p) = \frac{\langle s_{th} - \overline{s_{th}}, s_p - \overline{s_p} \rangle}{\|s_{th} - \overline{s_{th}}\| \cdot \|s_p - \overline{s_p}\|}$$

La signature retenue sera celle maximisant ce critère.

## 4 Cas des codes-barres *orientés*

Les codes-barres étudiés n'auront pas toujours leurs lignes parallèles aux bords de l'image. Le but de cette partie va être de trouver un angle que l'on utilisera pour effectuer une rotation sur l'image afin qu'elle devienne *droite*.

## 5 Traitement de la base de données des codes-barres