

Mise en œuvre de méthodes de recherche de lignes et d'analyse de textures pour la détection et la localisation de symboles bi-dimensionnels

Line detection and texture analysis for detection and localization of bi-dimensional symbols

B. Marcel M. Cattoen

Laboratoire d'Électronique de l'ENSEEIH
Groupe Signaux, Images et Communication

2, rue Charles Camichel
BP 7122
31071 Toulouse CEDEX 7
marcel@len7.enseeiht.fr

Résumé

Les symboles bi-dimensionnels représentent une nouvelle génération de codes symboliques dérivés des codes-à-barre, dont la lecture nécessite des méthodes par traitement d'image. Deux approches ont été envisagées pour la détection et la localisation des symboles : recherche de motifs constitutifs (lignes) et recherche des textures caractéristiques. Nous présentons dans cet article les différentes méthodes choisies et leurs résultats. Nous comparons ensuite ces résultats, et en tirons les conséquences pour obtenir une détection et une localisation plus efficace.

Mots clef

Analyse, imagerie, vision, détection de lignes, analyse de texture, codes symboliques.

Abstract

Bi-dimensional symbols represent new symbolic codes, derived from bar-codes, for which the reading process will be achieved by image processing methods. Two different means have been designed to process detection and localization: research of specific symbol's features (lines) or research of characteristic textures. We present here the chosen methods and their results. We then compare these results and propose a more efficient detection and localization algorithm.

Keywords

Analysis, imagery, vision, line detection, texture analysis, symbolic codes.

1 Les symboles bi-dimensionnels

Les symboles bi-dimensionnels (voir exemple figure 1) représentent une nouvelle génération de codes symboliques dérivés des codes-à-barre [10].

Leur conception à deux dimensions leur permet de coder beaucoup plus d'informations que les codes mono-dimensionnels, mais ne permet pas leur lecture par des lecteurs linéaires classiques (balayage de faisceau laser, "douchette" à base de barettes CCD...). Des méthodes utilisant le traitement d'image doivent donc être envisagées pour la détection, la localisation et la lecture ; ces méthodes doivent permettre d'obtenir les résultats de lecture suffisamment rapidement pour un usage ergonomique du dispositif.

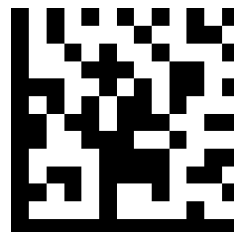


FIG. 1 - exemple de symbole Data Matrix

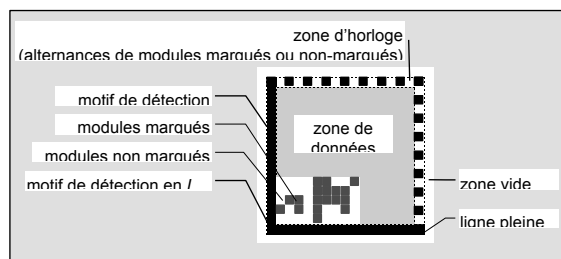


FIG. 2 - éléments d'un symbole Data Matrix

Les symboles sur lesquels portent cette étude sont de type Data-Matrix ([1], voir figure 1 et 2). Ils sont constitués d'une matrice de carrés noirs ou blancs (appelés *cellules*, *marquées* ou *non marquées*), encadrée par un motif de détection constitué de deux lignes pleines (appelé *motif de détection en L*) et d'une série de cellules de teinte alternante (appelée

zone d'horloge). Le nombre de cellules par lignes est appelé *fréquence*.

La méthode doit inclure la recherche du symbole (détection et localisation), puis sa lecture, à partir d'une image (voir figure 3).

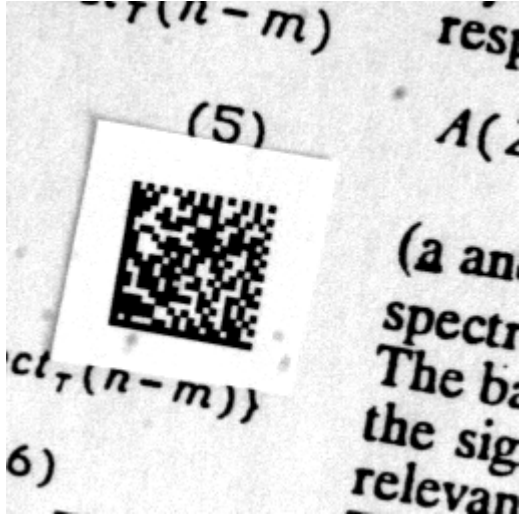


FIG. 3 - image d'un symbole *Data Matrix*

Nous présentons ici les étapes de détection et localisation, dont le but est de délivrer à l'étape suivante de lecture, avec le moins d'erreurs possible, les positions de symbole dans l'image. Différentes approches sont envisageables pour la détection et la localisation, notamment l'utilisation des motifs constitutifs des symboles (par exemple le motif de détection en L), et la recherche des textures caractéristiques d'un symbole dans l'image.

Ces deux approches ont été réalisées, chacune mettant en œuvre les méthodes qui semblaient les plus adaptées pour répondre au problème. Nous présentons ci-après les méthodes utilisées et leurs résultats ; puis nous les comparons et proposons une méthode faisant la synthèse des deux approches.

2 Méthode par détection de lignes

Cette méthode consiste à rechercher dans les images les grandes lignes susceptibles de représenter le motif de détection en L du symbole.

2.1 Choix de la méthode

La méthode par détection de ligne a été choisie en fonction des propriétés des images de symboles et des contraintes du dispositif à réaliser. Plusieurs méthodes sont présentes dans la littérature. La transformation de Hough ([7], [8], [11]) permet de réaliser des détections de lignes, mais leur localisation n'est pas complète, et la quantité de calculs nécessaire est élevée. Les simples calculs de gradient (Roberts, Sobel... voir [14] et [15]), pour

les étapes de détection de contours, ne donnent pas de résultats satisfaisants au point de vue de la localisation ; les filtres directionnels, dont on trouve des applications en robotique mobile pour les environnements structurés, s'appliquent difficilement dans notre cas où le symbole peut avoir une orientation quelconque. Des filtres plus complexes tels ceux de Canny [3] ou Haralick [9] nécessitent beaucoup de calculs pour un gain faible par rapport à des filtres de contours plus simples. Ces méthodes ont été écartées au profit de la méthode de Marr-Hildreth ([6], [9], [13]) qui consiste en la détection des passages à zéro du laplacien (méthode notée ZC^2 pour *Zero-Crossing of the second derivative* et détaillée dans [12]).

2.2 Propriétés générales de la méthode et ZC^2

Le laplacien (dérivée du gradient dans la direction du gradient) s'annule lorsque le gradient atteint un extremum. Les passages à zéro du laplacien représentent donc les points d'inflexion du signal-image. Tous les points d'inflexion dans un signal-image ne représentent pas des contours dans la scène [6] ; mais dans la zone où se trouve le symbole, l'image est bimodale, et ces points d'inflexion représentent en général les contours des éléments du symbole.

Le calcul du laplacien se fait par un masque de convolution. Le choix des paramètres de ce masque permet d'intégrer un lissage implicite du signal préalable à la dérivation ([17], [18]). Cette convolution peut être réalisée à partir d'opérateurs arithmétiques simples, de nature à permettre des traitements rapides.

La détection des pixels de contours est effectuée sans seuillage de l'intensité du passage à zéro. Les pixels détectés par erreur à cause de bruit ou de points d'inflexion hors contour sont éliminés dans une étape ultérieure. Cette absence de seuillage permet à la méthode de détecter les lignes de faible contraste et de ne pas nécessiter de paramétrage critique à ce niveau.

Le ZC^2 produit théoriquement des contours fins et fermés [16], ce qui simplifie les étapes suivantes de chaînage et de vectorisation. Dans le cas des images numériques, cette propriété n'est pas strictement vérifiée, mais elle s'applique relativement bien au cas des images bimodales à l'intérieur de la zone du symbole. Il ne manquera donc pas, ou très peu, de pixels de contour de symbole.



FIG. 4 - image d'un symbole Data Matrix

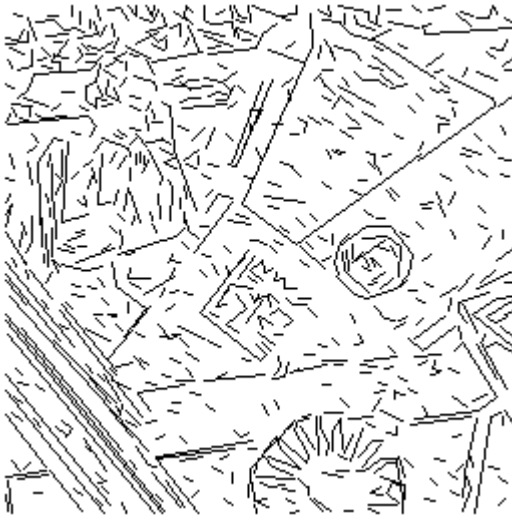


FIG. 5 - détection des lignes sur l'image de la figure 4

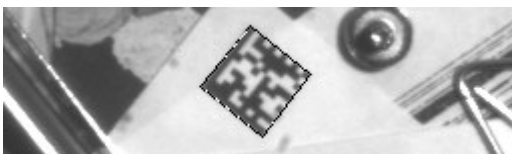


FIG. 6 - localisation du symbole

2.3 Chaînage et restauration de lignes

Des filtres directionnels calculent les directions de contours avec une précision faible ($22,5^\circ$ près) mais suffisante pour effectuer l'étape suivante de chaînage. En fonction de leur connexité et de leur direction de contour, les pixels de contour détectés sont assemblés en chaînes orientées. Les chaînes sont fragmentées en groupes de lignes orientées. Les pixels détectés par erreur seront éliminés lors de cette phase de chaînage, car seuls les vrais pixels de contours s'assemblent en chaînes de longueur conséquente.

Le ZC^2 présente l'inconvénient de déformer les contours au voisinage des coins aigus [2]. Certains pixels de contours peuvent aussi manquer en raison de bruit. Les lignes sont donc analysées et corrigées afin de joindre les lignes aux extrémités proches et obtenir ainsi une représentation plus cohérente de la scène par l'ensemble de lignes.

Une fois les lignes restaurées, l'algorithme effectue la sélection des candidats : il recherche, en commençant par les lignes les plus grandes, les couples de lignes susceptibles d'être le motif en L caractéristique du symbole (lignes orthogonales, de mêmes longueurs, dans l'orientation adéquate).

Les figures 4 et 5 montrent les résultats de détection de lignes. Le symbole présent dans l'image a été correctement détecté et localisé à partir des lignes comme on le voit figure 6.

3 Méthode par texture

L'approche par texture est basée sur le calcul multi-échelle d'un certain nombre de critères de texture dont la combinaison permet d'estimer la présence ou non de symbole. Pour couvrir les différentes fréquences spatiales que peut présenter un symbole, les critères sont calculés pour différentes échelles par une série d'échantillonnages préalables de l'image. On détermine la carte de présence probable de symbole pour chaque échelle séparément.

Plusieurs études existent dans la littérature sur l'analyse de texture, et proposent divers critères permettant de caractériser les scènes [4]. Ces critères peuvent être calculés directement à partir de l'image, ou à partir de formes dérivées (images de contours...). À partir des critères de [4], l'expérimentation nous a permis de développer trois critères mieux adaptés à notre application [5]. Les critères sélectionnés sont tous les trois calculés à partir de l'image de contour.

3.1 Analyse de texture

Les critères utilisés s'appliquent sur l'image de contours, pour des groupes de pixels appelés *texels*. Ils représentent la quantité dans chaque texel des pixels de contours ayant telle caractéristique. Certains critères sont des critères d'exclusion : leur valeur est faible dans les zones où un symbole est présent et élevée en dehors. Les figures 7 et 8 représentent des images de contour sur lesquelles ces critères sont appliqués.

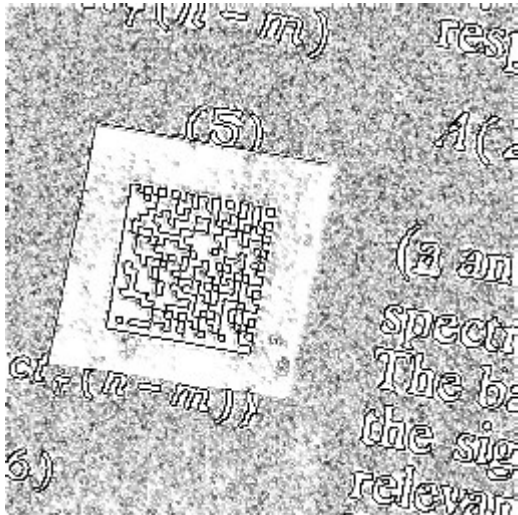


FIG. 7 - contours de l'image de la figure 3

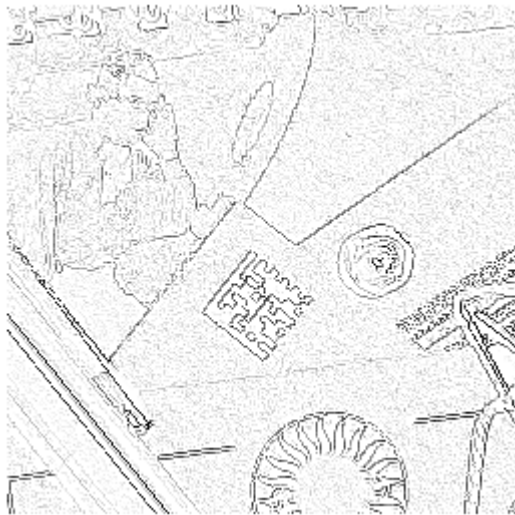


FIG. 8 - contours de l'image de la figure 4

Les critères utilisés sont la *continuité*, la *densité de contour*, et la *proximité*.

La connexité s'entend au sens du voisinage à 8, les texels sont de taille 8×8 . Les images sont de taille 256×256 , avec 256 niveaux de luminance. D'un point de vue algorithmique, ces critères ne sont pas calculés stricto sensu mais sont estimés, et l'approximation faite suffit pour tirer des conclusions sur la probabilité qu'à un texel d'appartenir à un symbole.

- Continuité C_o

Quantité de pixels des chaînes de longueur supérieure ou égale à 3. Ce critère a une valeur élevée dans les zones où se trouvent des contours, et donc là où se trouve un symbole. La limitation à trois pixels évite les détections dans les zones de bruit.

Pour chaque texel, on compte dans la carte de contours binaires le nombre de pixels isolés et ceux présents qui n'ont qu'un seul voisin, tels que ce voisin ait lui-même un seul voisin (qui est nécessairement le pixel que l'on compte). On retranche cette quantité du nombre de pixels de contours du texel.

- Densité de contours C_N

Quantité de pixels de contour agglomérés en blocs de taille 2×2 ou plus. C'est un critère d'exclusion : les symboles ne présentant pas de traits fins, il ne devrait pas y avoir de contours agglomérés (avec une carte de contours minces calculée par un $ZC\sigma^2$).

Pour chaque texel, on compte dans la carte de contours binaires le nombre de pixels présents ayant des voisins dans les trois directions Est, Nord-Est et Nord. Ce calcul représente moins le nombre de pixels agglomérés que le nombre de blocs : un seul pixel est compté pour un bloc isolé de taille 2×2 (les blocs sont rarement plus grands étant donné que nous utilisons des contours fins), et quelle que soit la forme du bloc, certains pixels du bord du bloc ne sont pas comptés.

- Proximité C_p

Quantité de pixels rapprochés. C'est un critère d'exclusion : les symboles présentent des contours relativement éloignés les uns des autres.

On compte dans la carte de contours binaires les pixels de non-contour ayant deux contours dans les directions Est et Ouest ou dans les directions Nord et Sud. Ceux qui ont quatre contours dans les directions Est, Nord, Ouest et Sud comptent double.

3.2 Carte de détection

Les cartes de critère sont combinées en une carte unique où chaque texel marqué correspond à une présence probable de symbole en cet endroit. La figure 10 représente la carte de présence probable de l'image de la figure 9 (nous avons reproduit les luminances de l'image sur les texels marqués).

Une combinaison linéaire des critères, suivie d'un seuillage, permettrait une grande souplesse dans la pondération des critères, mais le choix des coefficients serait difficile à réaliser. Une méthode équivalente a été préférée qui consiste à seuiller d'abord séparément chaque critère, puis à en faire la combinaison booléenne. Cette méthode permet de choisir séparément, et avant combinaison, le seuil à appliquer sur un critère en fonction de la qualité et de l'amplitude de sa réponse. Les valeurs des seuils ont été déterminées par une méthode heuristique, en

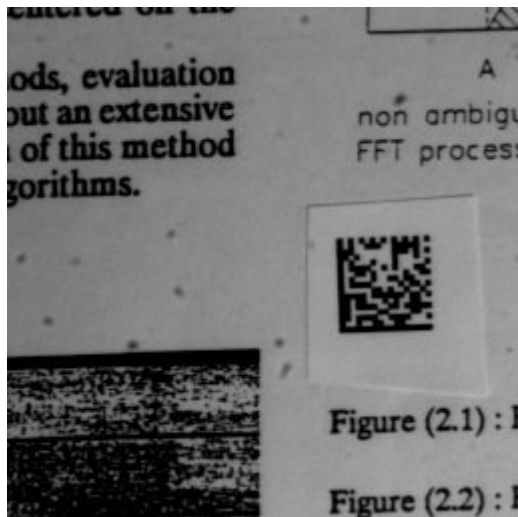


FIG. 9 - image d'un symbole

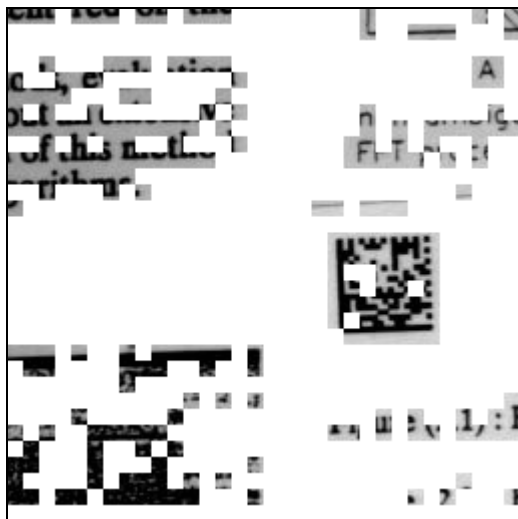


FIG. 10 - carte de présence probable pour l'image de la figure 9

faisant des essais sur divers types d'images et de symboles.

3.3 Localisation

S'il y a un symbole dans l'image, les texels correspondant à cette zone dans la carte de présence probable sont en majorité marqués. Puisque la zone où se trouve un symbole est carrée, le groupe de texel marqué et correspondant à cette zone doit avoir une forme convexe proche d'un carré. Un traitement morphologique (ouverture et fermeture) y est appliqué afin d'éliminer les texels isolés.

Les texels restants sont regroupés en objets (groupes connexes) dont on mesure les caractéristiques géométriques. Les groupes dont l'excentricité est trop élevée, ou dont la surface est trop petite, sont

rejetés. Les contours des groupes restants sont extraits et analysés afin de déterminer la position exacte du carré de la zone du symbole.

L'ensemble des calculs de contours, analyse de texture et localisation, est effectué pour chaque échelle. Les différentes échelles sont en inverse de puissances de 2 ($1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8} \dots$). Chaque changement d'échelle équivaut à doubler la taille du côté du texel. La dernière échelle ($\frac{1}{32}$) est celle pour laquelle la taille du texel équivalent atteint celle de l'image.

4 Comparaison des résultats et combinaison des méthodes

Les deux méthodes appliquées séparément permettent dans l'ensemble de détecter les symboles, chacune présentant néanmoins des faiblesses. Des exemples sur des images de symbole permettent de le constater.

4.1 Résultats de détection de symboles par lignes et textures

4.1.1 Méthode par détection de lignes

La méthode par lignes sélectionne tous les couples de lignes ayant les caractéristiques du motif en L . Certains de ces couples peuvent ne pas être le bord d'un symbole, surtout avec les types d'images structurées dans lesquels se trouvent les symboles.

Les symboles des figures 3, 4 et 9 sont uniques dans l'image, mais dans la figure 11, un autre candidat est proposé à la lecture. La méthode n'a aucun moyen de trancher en faveur de l'un ou l'autre candidat et les deux doivent être proposés à la lecture (voir figure 12).

4.1.2 Méthode par analyse de texture

La méthode par texture est efficace en détection, mais peu précise en localisation. La nature de la texture (et donc la présence ou pas d'un symbole) est établie sur des texels de 8 pixels de côté ; ces pixels, selon l'échelle, peuvent à leur tour représenter plusieurs pixels de l'image d'origine. Les position et localisation du symbole doivent être connues avec exactitude afin de mener à bien les étapes suivantes de lecture. Si la position du symbole est erronée, il est fort probable que le système soit incapable de déterminer la fréquence du symbole, et *a fortiori* d'en faire un échantillonnage correct. Les symboles des figures 3, 4 et 9 sont convenablement localisés, mais ce n'est pas le cas de celui de la figure 13.

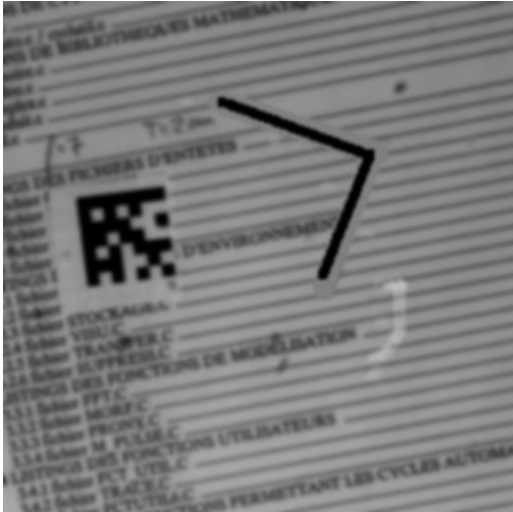


FIG. 11 - image d'un symbole avec motif perturbateur



FIG. 12 - candidats retenus

4.2 Combinaison des méthodes

Nous voyons que les défauts de chaque méthode sont assez différents. La méthode par ligne donne de bons résultats mais peut-être faussée par certains éléments de l'image. La méthode par texture réalise la détection, mais n'est pas fiable pour la localisation.

Nous avons donc conçu une méthode qui réalise la

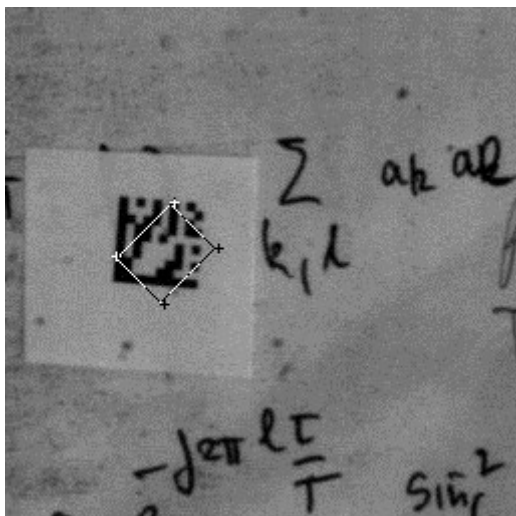


FIG. 13 - détection d'un symbole avec échec de la localisation

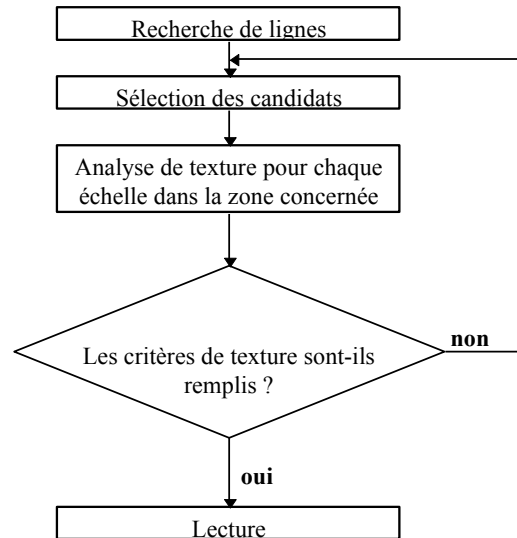


FIG. 14 - organigramme de la méthode générale

détection et la localisation, en s'appuyant sur les performances positives de chacune des approches. Elle procède de la façon suivante.

La méthode par recherche de lignes est tout d'abord appliquée. L'ensemble de lignes délivrées est analysé afin d'en tirer un certain nombre de candidats ayant les propriétés du motif de détection en L . Pour chacun de ces couples de lignes, nous analysons sur plusieurs échelles la texture à l'intérieur de la zone délimitée ; les couples sont rejetés si la proportion de texels marqués est trop faible dans cette zone. De cette manière, les échecs décrits à la figure 12 ne se produisent plus. L'étape de localisation dans la méthode par texture, qui constituait sa principale faiblesse, n'est plus réalisée.

L'enchaînement des opérations est présenté figure 14.

Conclusion

La combinaison des deux méthodes présentées permet de réussir des détections là où l'une ou l'autre méthode, prise séparément, aurait échoué. La quantité de calculs est optimisée (l'image de contours n'est calculée qu'une seule fois ; les caractéristiques de textures ne sont pas calculées pour toute l'image). La validation des détections de symboles par l'analyse de texture permet d'éviter de tenter des opérations de lecture vouées à l'échec qui ralentiraient le fonctionnement du dispositif.

Les faiblesses d'une méthode sont contournées par la participation de l'autre. Les points forts de chaque méthode sont conservés.

L'approche initiale avait été de chercher quelle propriété intrinsèque des images de symbole pourrait donner de meilleurs résultats. Il s'est avéré que

l'utilisation conjointe de plusieurs propriétés (motif en L et texture) était nécessaire pour obtenir de bons résultats de détection et localisation.

La méthode ainsi obtenue a été testée. Elle est assez sensible au bruit (qui perturbe la détection de ligne), mais permet la détection et la localisation même sous un faible contraste (grâce à l'utilisation de tous les pixels de contour, y compris de faible intensité).

Bibliographie

- [1] *Uniform symbology specification-Data Matrix - Draft*, AIM USA Technical Specification, AIM^{USA} Technology Group, TC-TSC/96-006, 12 décembre 1995.
- [2] V. Berzins, Accuracy of laplacian edge detectors, *Computer Vision, Graphics, and Image Processing*, Vol. 27, 1984, p. 195-210.
- [3] J. F. Canny, A computational approach to edge detection, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, n°6, 1986, p. 679-698.
- [4] A. Casals, J. Amat et A. Grau, Texture parametrization method for image segmentation, Second European Conference on Computer Vision, Santa Margherita Ligure, Italie, 1992, p. 160-164.
- [5] S. Délestaing, Détection et localisation de codes bi-dimensionnels par analyse de texture, rapport de DEA-SIC GTTSI, Laboratoire d'Électronique de l'ENSEEIH, INPT, Toulouse, 9 septembre 1996
- [6] M. M. Fleck, Some defects in finite-difference edge finders, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 14, n°3, mars 1992, p. 337-345.
- [7] G. Foresti, V. Murino, C. S. Regazzoni et G. Vernazza, Grouping of rectilinear segments by the labeled Hough transform, *CVGIP : Image Understanding*, Vol. 58, n°3, novembre 1994, p. 22-42.
- [8] A. K. Gupta, S. Chaudhury et G. Parthasarathy, A new approach for aggregating edge points into line segments, *Pattern Recognition*, Vol. 26, n°7, 1993, p. 1069-1086.
- [9] R. M. Haralick, Digital step edges from zero crossing of second directional derivatives, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 6, n°1, janvier 1984, p. 58-68.
- [10] C. K. Harmon, Cures for the common code, IDSystems, avril 1993, p. 24-28.
- [11] V. F. Leavers, Which Hough transform ?, *CVGIP : Image Understanding*, Vol. 58, n°2, septembre 1993, p. 250-264.
- [12] B. Marcel et M. Cattoen, Edge and line detection in low level analysis, *Third Workshop on Electronic Control and Measuring Systems*, Université Paul Sabatier, Toulouse, 2-3 juin 1997, p. 89-97
- [13] D. Marr et E. Hildreth, Theory of edge detection, *Proc. Royal Society of London*, 1980, pp.187-217.
- [14] V. S. Nalwa et T. O. Binford, On detecting edges, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, n°6, nov. 1986, p. 699-714.
- [15] T. Peli et D. Malah, A study of edge detection, *Computer Graphics and Image Processing*, Vol. 20, 1982, p. 1-21.
- [16] V. Torre et T. A. Poggio, On edge detection, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, n°2, mars 1986, p. 147-163.
- [17] M.-J. J. Wang, S.-C. Chang, C.-M. Liu et W.-Y. Wu, A new edge detection method through template matching, *International journal of Pattern Recognition and Artificial Intelligence*, Vol. 8, n°4, 1994, p. 899-917.
- [18] J. M. White et G. D. Rohrer, Image thresholding for optical character recognition and other applications requiring character image extraction, *IBM J. res. develop.*, Vol. 27, n°4, juillet 1983, p. 400-411.