DOSSIER

Mathématiques appliquées

Les codes à barres 2D

Bruno Rolland est Directeur de Axicon France, experts en contrôle qualité des codes à barres, www.axicon.fr

Dans l'édition précédente de MV (Numéro 12) nous avons présenté les codes à barres en nous concentrant sur les codes à barres dits « linéaires » ou « 1D » par opposition aux codes à barres matriciels ou bidimensionnels, « 2D ». Notons qu'Il existe aussi des codes à barres hybrides, multilignes ou empilés ... Dans cet article nous nous pencherons sur les codes 2D matriciels comme le Data Matrix ou le code QR.

1D versus 2D

La différence fondamentale entre les deux types se trouve dans la manière de les lire. Pour un code 1D, il suffit qu'une ligne de scan traverse le code de la première à la dernière barre, perpendiculairement aux barres ou pas (voir Monsieur Thalès pour la justification). Pour un code 2D, il faut prendre une image de la totalité du code et analyser ensuite cette image pour décoder le code.

Data Matrix

Il y a d'autres différences notoires, la première étant la capacité des codes 2D à encoder plus d'information dans un espace plus réduit. Les codes 2D sont beaucoup plus denses et offrent une alternative aux codes 1D tant les contraintes liées au manque de place sur les emballages sont importantes.

QR Code

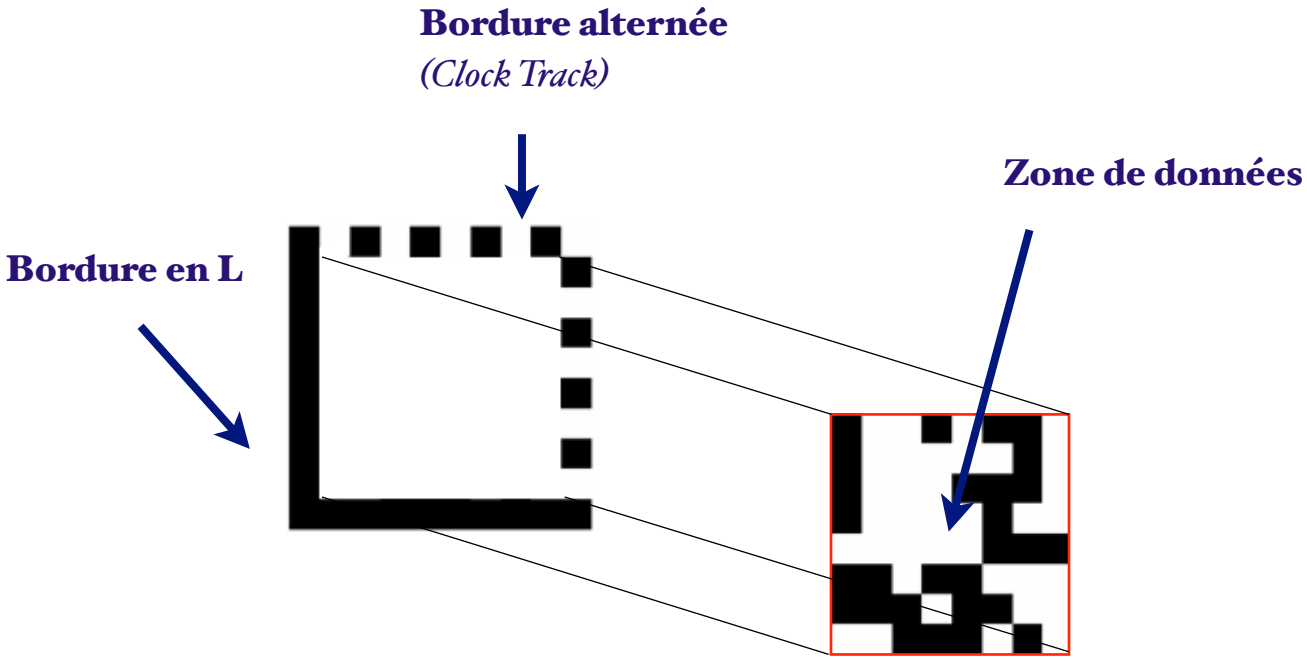
Le choix entre code 1D ou code 2D est généralement imposé par des réglementations, par des contraintes technologiques ou par des usages.

The Matrix

Pour le QR (Quick Response) code ou pour le Data Matrix (images ci-dessus), comme le nom de ce dernier l'indique on se trouve devant un code matriciel. Généralement N lignes et N colonnes (On rencontre parfois mais assez rarement des code Data Matrix au format N x M). Dans l'exemple ci-contre le Data Matrix est un 10 x 10.

A l'intersection des lignes et des colonnes se trouvent les modules, les petits carrés noir ou blanc qui constituent le Data Matrix.

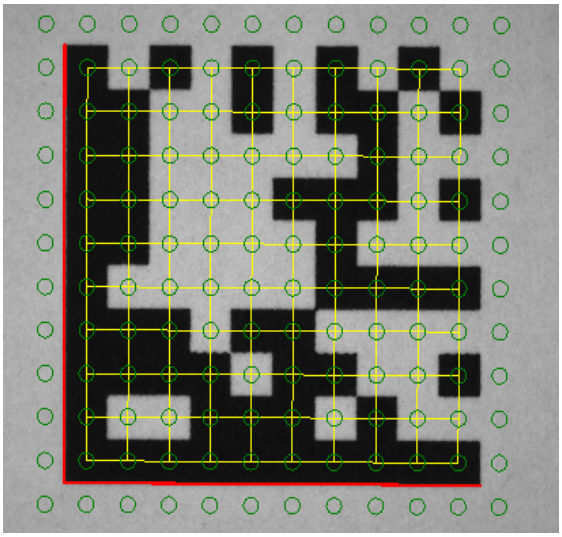
La bordure extérieure du code est immuable, composée de la bordure en L et de la bordure alternée, qu'on retrouvera donc dans chaque Data Matrix. La matrice intérieure correspond à la zone des données et sa taille de matrice est .... N-2 x N-2 bien sûr. 8 x 8 dans cet exemple.



Les étapes du décodage

Le décodage d'un Data Matrix se fait en plusieurs étapes.

La première consiste à prendre une « photo » du code, ensuite l'algorithme de décodage se met en route. On localise d'abord la bordure en L, deux segments qui se rejoignent à angle droit.



On localise ensuite les modules de la bordure alternée, puis on trace des droites passant par les centres de ces modules et parallèles à la branche de la bordure en L adjacente.

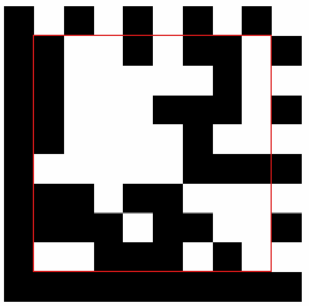
On trace ainsi la grille de lecture du code (en jaune sur ce graphique) dont les points d'intersection sont centrés sur les modules du Data Matrix

Maintenant que la grille de lecture a été tracée, on peut passer au décodage de la zone des données, et le principe est très simple. Il faut déterminer pour chaque module si il est de couleur claire ou sombre.

Notez qu'on n'a pas toujours des Data Matrix ou des QR imprimés en encre noire sur un fond blanc comme dans cet exemple. Donc on fait référence généralement à des modules clairs et sombres. Ce qui est important par contre c'est d'avoir un contraste suffisant entre les deux types de modules.

Aux modules sombres on donnera la valeur '1' et au module clairs la valeur '0'. Ces valeurs binaires sont regroupées et ordonnées par groupes de huit (les codewords) selon un découpage particulier spécifié dans le standard ISO du Data Matrix (pour référence ISO/IEC 16022). Regardons ensemble l'exemple du code Data Matrix représentant les données 123456.

Le Data Matrix permet d'encoder les chiffres par paires, donc on utilise les 3 premiers codewords pour encoder successivement 12, 34 et 56. On pourrait aussi bien sûr encoder les 6 chiffres séparément mais cela mobiliserait six codewords au lieu de trois et nous ferait perdre un peu de place. Autant optimiser l'encodage ! Ils nous reste encore 5 octets qui vont être utilisés pour ajouter des jokers... plus précisément des codewords de correction d'erreur. On y reviendra.



1	0	0	1	0	1	1	0
1	0	0	0	0	0	1	0
1	0	0	0	1	1	1	0
1	0	0	0	0	1	0	0
0	0	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
0	0	1	1	1	0	1	0

Découpage et position des codewords d'une matrice 8 x 8.
2.1 = premier module du second codeword

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6

1	0	0	1	0	1	1	0
1	0	0	0	0	0	1	0
1	0	0	0	1	1	1	0
1	0	0	0	0	1	0	0
0	0	0	0	0	1	1	1
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
0	0	1	1	1	0	1	0

10001110 – 10100100 – 10111010 – 01110010 – 00011001 – 00000101 – 01011000 - 01100110

123456 puis 5 codewords de correction d' erreur

Source : Guide GS1 Data Matrix ECC 200

Prenons le premier codeword de données: 10001110

On convertir la valeur binaire du codeword en valeur décimale.

Pour la valeur binaire 10001110 cela donne: 1x27+0x26+0x25+0x24+1x23+1x22+1x21+0x20 = 142

On se conforme ensuite à la table ci-dessous pour déterminer la valeur de la donnée: Pour un codeword entre 130 et 229, on soustrait 130 à la valeur décimale 142 – 130 = 12

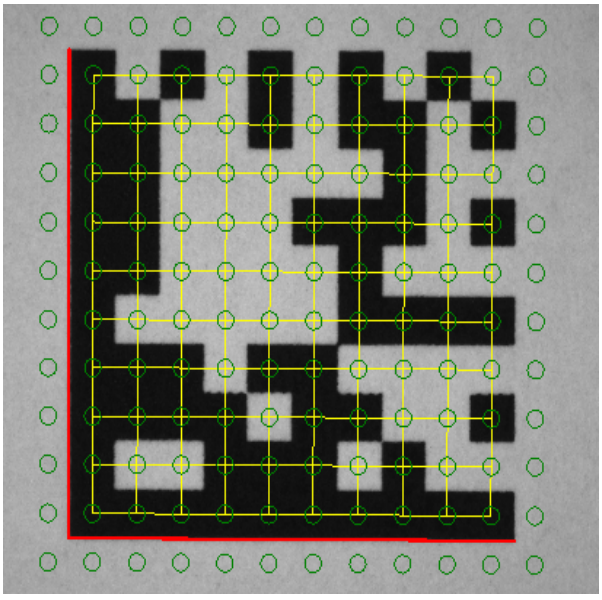
Un petit exercice: Faites de même avec les codewords 10100100 et 10111010 pour retrouver les données 34 et 56.

Codeword	Data or function
1-128	ASCII data (ASCII value + 1)
129	Pad
130-229	2-digit data 00-99 (Numeric Value + 130)
230	Latch to C40 encodation
231	Latch to Base 256 encodation
232	FNC1
233	Structured Append
234	Reader Programming
235	Upper Shrit (shift to Extended ASCII)
236	05 Macro
237	06 Macro
238	Latch to ANSI X12 encodation
239	Latch to Text encodation
240	Latch to EDIFACT encodation
241	ECI Character
242-255	Not to be used in ASCII encodation

Les codewords de 1 à 128 permettent d'encoder les caractères de la table ASCII, on utilise les codewords de 130 à 229 pour encoder les 100 paires de chiffres de 00 à 99.

Comment le lecteur détermine la valeur binaire d'un module

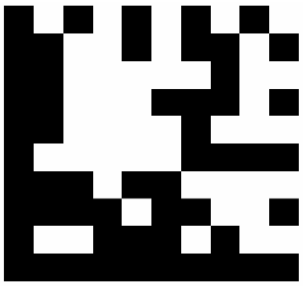
Un détail dont nous n'avons toujours pas parlé est celui de savoir comment le lecteur détermine si un module doit avoir la valeur 1 ou la valeur 0, si il est sombre ou clair. Evidemment les lecteurs modernes utilisent des technologies diverses et très performantes pour décoder en quelques centièmes de secondes mais schématiquement, la démarche est assez simple. Le capteur photo sensible du lecteur optique fonctionne (majoritairement) sous une source de lumière rouge (autour de 670 nanomètres) et donc ne distingue plus des couleurs mais des nuances de gris. Si vous ne me croyez pas essayez de mettre des lunettes à filtres rouges et vous comprendrez ou alors demandez au prof de physique de vous expliquer....



A chaque point d'intersection de la grille de lecture, donc pour chaque module de la matrice, on trace un cercle dont le diamètre couvre entre 50% et 80% de la largeur du module. Ce cercle est la zone d'échantillonnage pour déterminer la nature binaire du module en question. Le capteur mesure le niveau de gris de chaque pixel de l'image qui se trouve à l'intérieur du cercle et calcule pour chaque module sa valeur moyenne (exprimée en pourcentage) sur l'échelle des niveaux de gris. Les modules sombres auront une valeur basse et les modules clairs une valeur élevée. On prend la valeur maximale et la valeur minimale trouvées sur l'ensemble des modules de la matrice, on détermine ensuite un seuil à mi chemin entre la valeur maximale et minimale. et la règle de décision est simple, à tous les modules dont la valeur moyenne de niveau de gris est supérieure au seuil on donne la valeur 0 et à tous ceux qui sont au dessous du seuil on donne la valeur 1 !

La correction d'erreur

A ce stade de notre article, vous commencez à avoir un peu plus de recul pour comprendre ce qui se cache derrière ces symboles et même si nous nous sommes surtout concentrés sur le code Data Matrix, le code QR est équivalent à bien des égards. Quelque chose devrait pourtant vous préoccuper... Que se passe-t-il si par accident un seul petit module blanc, à cause d'une tache d'encre par exemple, était interprété comme un module sombre, ce qui doit être lu comme un 0 est lu comme un 1, la valeur binaire et décimale du codeword change complètement et la donnée est fausse ... Si on reprend l'exemple du code 123456. Le module blanc qui se trouve en deuxième position du premier codeword (position 1.2 dans la lecture d'un 8 x 8) est tâché et le lecteur va lui associer la valeur 1 au lieu de 0



Code original
Premier codeword: 10100100
Valeur décimale: 142
Valeur de la première donnée: 12
Encodage: 123456v

Code endommagé
Premier codeword: 11100100
Valeur décimale: 228
Valeur de la première donnée: 98
Encodage 983456

Source : Google images

C'est là qu'intervient la correction d'erreur. Comme leur nom l'indique plutôt clairement, les codewords de correction d'erreur, qui viennent compléter la matrice en sus des codewords de données, servent à réparer d'éventuelles erreurs de lecture ou défauts d'impression et corriger des modules dont la valeur aurait été mal interprétée par le lecteur. Un des cas les plus concrets est celui des divers logos et graphismes qui semblent envahir les codes QR des annonceurs, soucieux de représenter leur marque jusque dans le code QR. Un logo en plein milieu d'un code QR va modifier la nature des modules qu'il couvre et qu'il occulte.

Plus on va avoir d'erreurs à corriger plus on va solliciter ces codes de correction d'erreur et leur capacité à corriger n'est pas infinie. Si le nombre de codewords endommagés dépasse la capacité du code à se corriger alors le code ne pourra pas être lu.

Je vais désormais laisser la plume à mon camarade Marc Trichard, qui va vous expliquer les algorithmes de Solomon Reed sur lesquels repose la miraculeuse correction d'erreur !

Bruno Rolland

Un petit anagramme…

L'anagramme à trouver est celui de : Léonard de Vinci

Source : Anagrammes renversantes, E. Klein, J. Perry-Salkow, Flammarion.

Réponse : le don divin créa