

SAV

Traitement d'image

Romuald Mosqueron

Novembre 2017



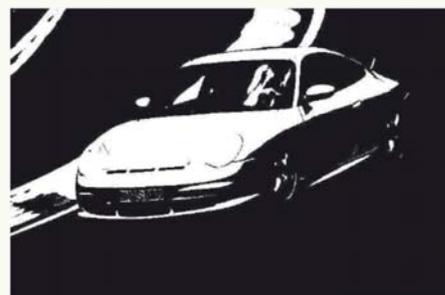
26/11/2017

Binarisation

Valeur de l'image binarisée=1 si Valeur pixel > Seuil
sinon =0

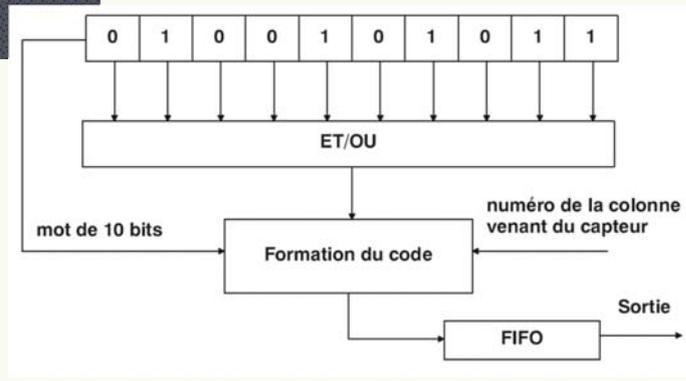
Différents types de binarisation

- Global : Un seuil pour toute l'image
- Local : Seuil dépendant des propriétés locale du point
- Local adaptatif : Position en plus

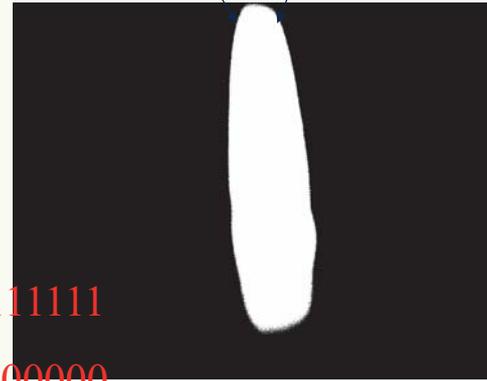


26/11/2017

Binarisation



Colonne 12 Colonne 25
Valeur Valeur
0000111111 1111000000



Seuillage des 10 pixels en parallèle

Codage de l'objet

Codage => n° colonne + mot 10 bits

00000000 & 00001100 & 000000 & 0000111111

00000000 & 00011001 & 000000 & 1111000000

Détection de contours

Filtre de Sobel :

$$g(x, y) = |(g_1(x, y))| + |(g_2(x, y))|$$

$$h_1 =$$

-1	0	1
-2	0	2
-1	0	1

$$g_1(x, y) = (f(x, y) * h_1) \text{ et } g_2(x, y) = (f(x, y) * h_2)$$

Transformée en Z

$$G_1(z) = (F(z) + 2Z^{-1}F(Z) + Z^{-2}F(Z)) - Z^{-2N}(F(z) + 2Z^{-1}F(Z) + Z^{-2}F(Z))$$

$$G_2(z) = (F(z) + 2Z^{-N}F(Z) + Z^{-2N}F(Z)) - Z^{-2}(F(z) + 2Z^{-N}F(Z) + Z^{-2N}F(Z))$$

Après simplification

$$G_1(z) = F(z) \cdot (1 - Z^{-2N}) \cdot (1 + Z^{-1}) \cdot (1 - Z^{-1})$$

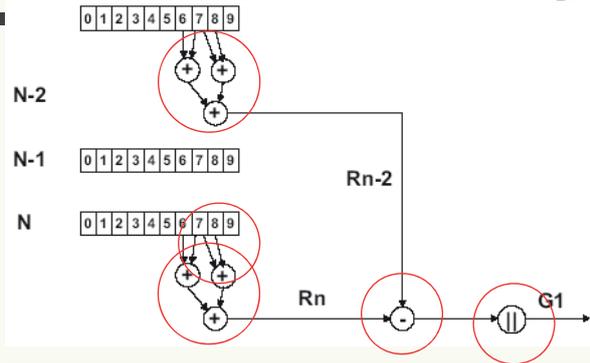
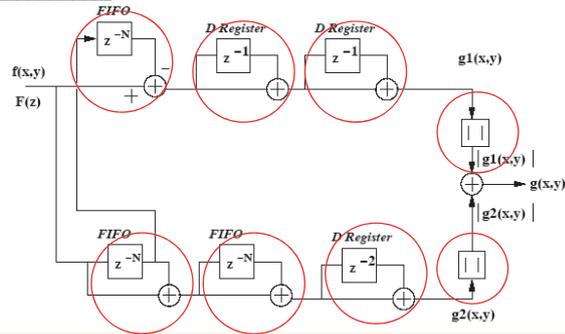
$$h_2 =$$

-1	-2	-1
0	0	0
1	2	1

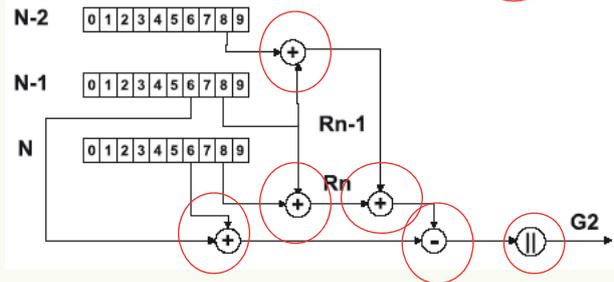
$$G_2(z) = F(z) \cdot (1 + Z^{-N}) \cdot (1 + Z^{-N}) \cdot (1 - Z^{-2})$$

Simple filtre numérique du 1^{er} ordre

Détection de contours



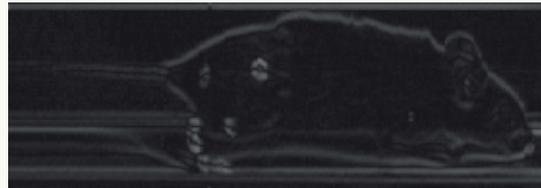
- 3 FIFOs de 10 pixels
- 10 pixels en parallèle
- Évite les redondances d'opérations
- Économie de registre



26/11/2017

6

Détection de contours



- 1560 slices
- 9 BRAMs
- 500 images/s
- Utilisation possible avant d'autres traitement

26/11/2017

7

Barycentre

$$x_G = \frac{1}{a_1 + a_2 + \dots + a_n} (a_1 x_{A_1} + a_2 x_{A_2} + \dots + a_n x_{A_n})$$

- Calcul sur la ligne uniquement (1D)
- Sortie du numérateur et du dénominateur
- Division et calcul 2D sur l'ordinateur

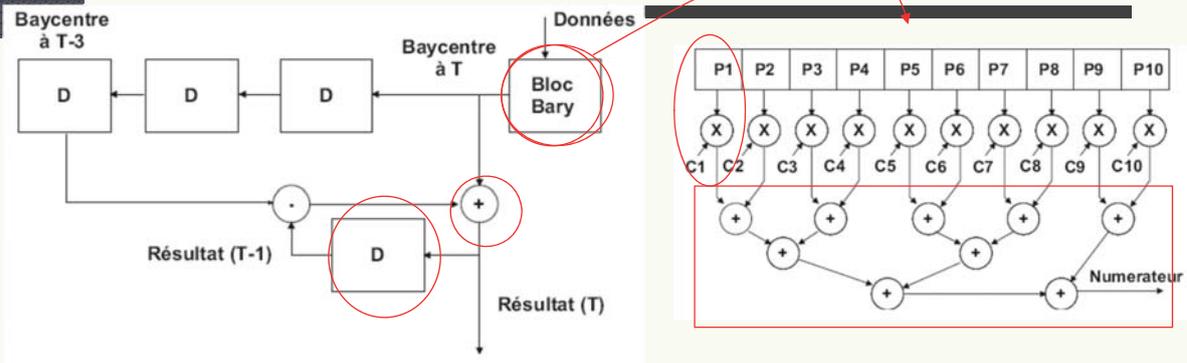


- Fenêtre multiple de 10 pixels
- Fenêtre de 30 pixels par pas de 10 pixels
- Calcul de 1 barycentre de 10 points par fenêtre de 30

26/11/2017

8

Barycentre



- Numérateur sur 41 bits et dénominateur sur 15 bits

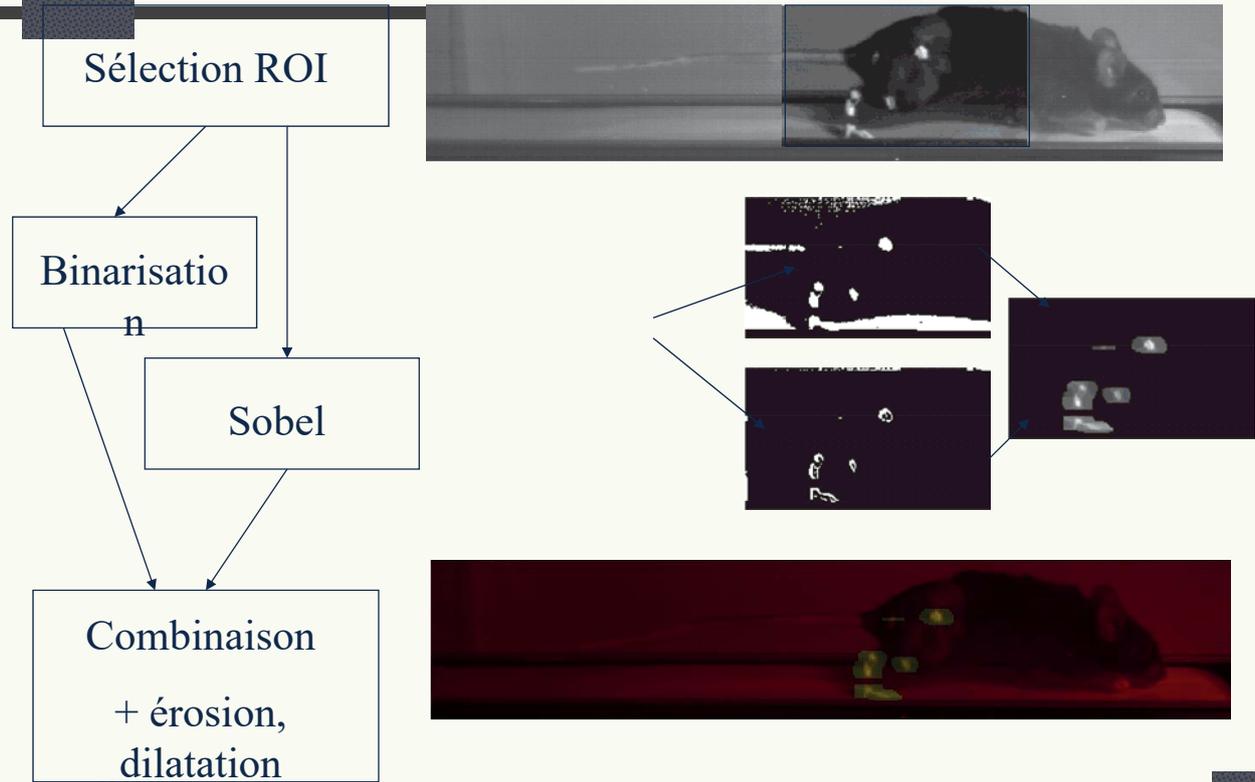
=> 8 octets

- 500 images/s
- 636 slices et 0 BRAM
- Limité à 64 calcul de barycentre par ligne

26/11/2017

9

Extractions de marqueurs



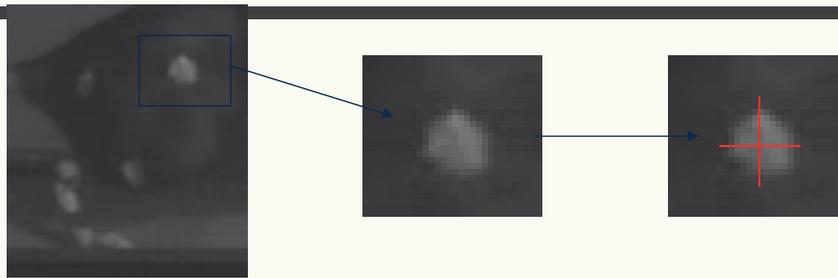
26/11/2017

Extractions de marqueurs

$$erosion(x; y) = \text{minimum}(f(x; y); f(x_{i-1}; y); \dots; f(x_{i+2}; y_{i+2}))$$

$$dilation(x; y) = \text{maximum}(f(x; y); f(x_{i-1}; y); \dots; f(x_{i+2}; y_{i+2}))$$

Extractions de marqueurs

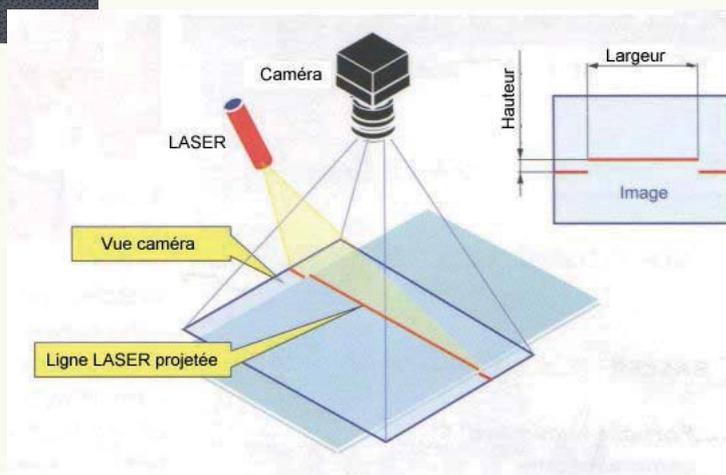


- Sortie des coordonnées du marqueur sur le même principe que le barycentre
- Implantation validée mais pas implantée
- ~3000 slices et 9 BRAMs
- 500 images/s

26/11/2017

12

Profilométrie



Calcul du barycentre
du trait laser sur
chaque ligne

Même principe que le
barycentre

1 par ligne

- 500000 profils/s
- Profilométrie haute vitesse
- 751 slices et 0 BRAM

26/11/2017

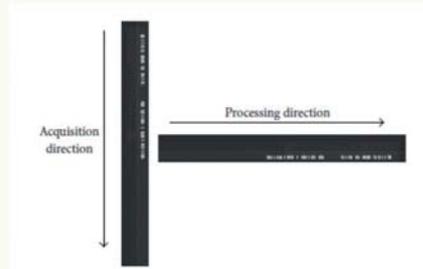
13



(a) Area scan mode



(b) Line scan mode



(a) Original image



(b) Image after high pass filtering



(c) Image after dilation



(d) Image after subsampling



(e) Image after blobbing



(a) Zoom of the subsampling image on one bar code area



(b) Zoom on the same bar code area after the blobbing



FIGURE 12: Zone transferred after processing to read the bar code.



(a) Original image not readable at sight



(b) Image readable after processing

26/11/2017

Processing	Time (ms)
Acquisition	15.4
Transfer between the 2 FPGAs	4.6
Transposition	1.54
High pass filtering	1.54
Dilatation plus subsampling	1.54
Transfer in the processor of the subsampling image	0.15
Blobbing	4
Transfer in the processor of the bar code image	0.11
Reading the bar code image	12
Total	40.88

	Sequential	Parallel	PC (ROI)
Time processing (ms)	40	40	40
Number of letter	15	30	15
Theoretical speed (m/s)	4	8	4



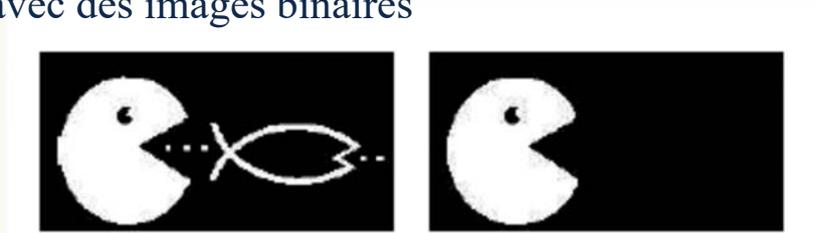
(a) Coprocesor platform plus camera



(b) PC and coprocesor platform

26/11/2017

Plus facile avec des images binaires



Suppression des structures fines



Remplissage des trous

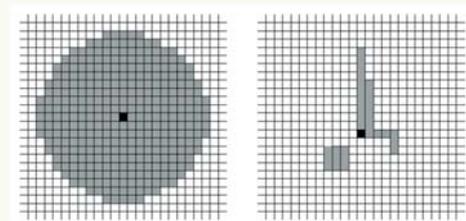
26/11/2017

16

Element structurant

Un élément structurant est composé :

- d'un pixel central (en noir)
- d'un ensemble de pixels (en gris)



Les algorithmes de morphologie mathématique parcourent l'image :

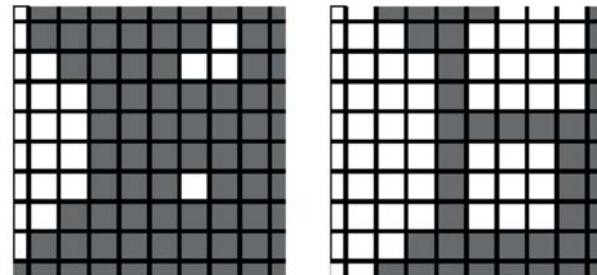
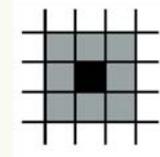
- en chaque pixel de départ $(u(i;j))$ on place l'élément structurant centré sur le pixel noir,
- un test est réalisé pour déterminer la couleur du pixel d'arrivée $u'(i;j)$.

26/11/2017

17

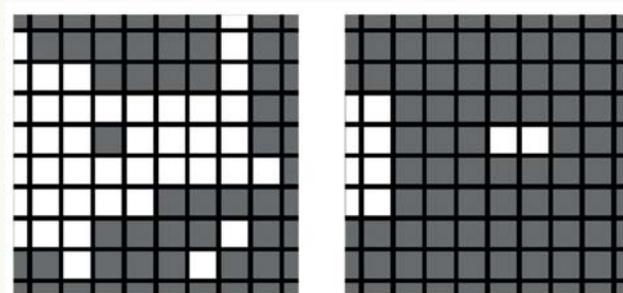
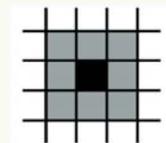
Parcourir les pixels de l'image u , pour chaque pixel $u(i;j)$:

1. Centrer l'élément structurant sur ce pixel,
2. Considérer les voisins du pixel dans l'élément structurant (gris + noir)
3. Si l'un de ces pixels est en blanc, mettre $u'(i;j)$ en blanc.

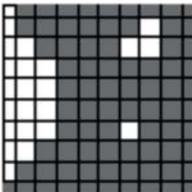
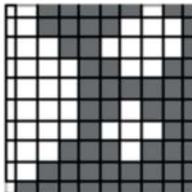
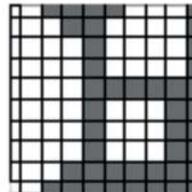


Parcourir les pixels de l'image u , pour chaque pixel $u(i;j)$:

1. Centrer l'élément structurant sur ce pixel,
2. Considérer les voisins du pixel (gris + noir de l'élément structurant)
3. Si l'un de ces pixels est en noir, mettre $u'(i;j)$ en noir.



Exemples

	Image originale		
Dilatation			
Érosion	