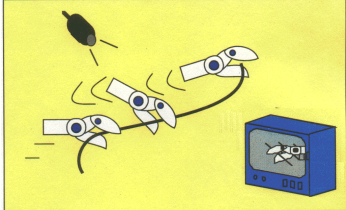


Cours de Robotique et d'Automatisation Facultés de Technologies Industrielles (TIN),
et Formation En Emploi (FEE)
Filières microtechniques et électronique et automatisation industrielle (MI et EAI)
LaRA - Laboratoire de Robotique et Automatisation

Vision par ordinateur – Traitement 2D

(Partie 4.4. du cours, point 3. Et manipulation No 24)



HESSO-HEIG-VD,
J.-D. Dessimoz,
29.11.2016

heig-VD
HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-VD.ch

institut d'Automatisation Industrielle
LaRA
Laboratoire de Robotique et Automatisation

Hes·SO
Haute Ecole Spécialisée
de Suisse occidentale

Vision par ordinateur

• 1 Introduction	1
• 2 Acquisition d'images	5
• 3 Traitement bidimensionnel d'images	27
• 4 Analyse de scènes	48
• 5 Particularités de la vision pour robots	68
• 6 Exemples d'application	
• 7 Conclusion	

Robotique et automatisation, JDZ/JDD 20.04.2011 2

3 Traitement bidimensionnel d'images


• 3.1 Opérations par pixels individuels (point par point)	28
• 3.2 Modifications des pixels d'après leur voisinage	29
• 3.3 Opérations globales ou par régions	39
• 3.4 Détection de mouvement	45
• 3.5 Transformée en cosinus	46
• 3.6 Temps de traitement	46

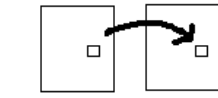
Robotique et automatisation, JDZ/JDD 20.04.2011 3

Traitement de pixel pour affichage en pseudocouleurs

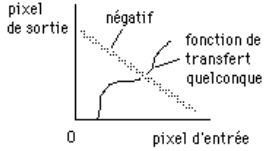
Implémentation:

- « palette » (tableau - array)
- « look-up table » (petite mémoire)





Vision 10



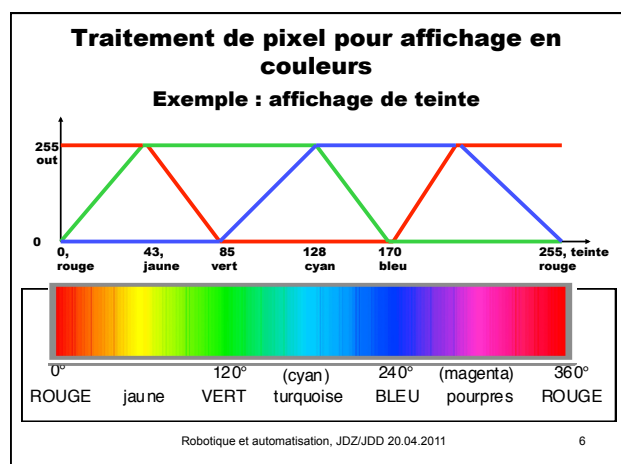
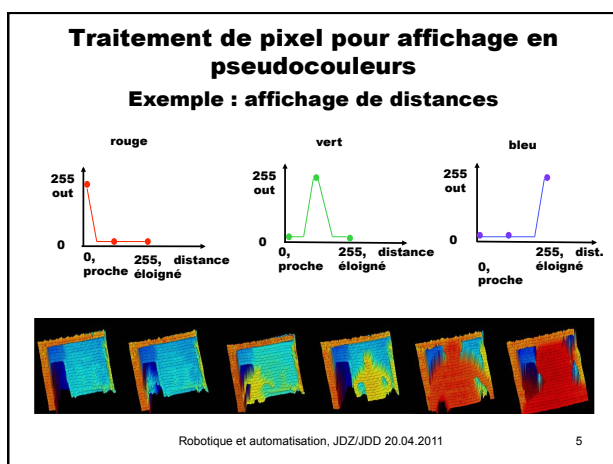
pixel de sortie

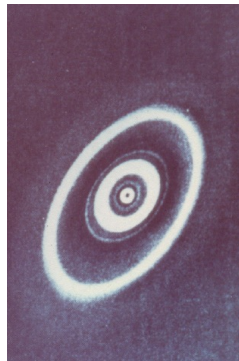
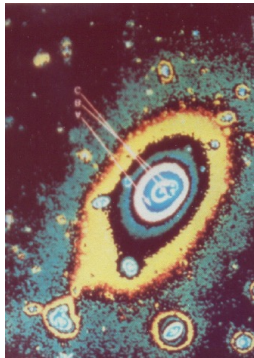
0 pixel d'entrée

fonction de transfert quelconque

negatif

Robotique et automatisation, JDZ/JDD 20.04.2011 4

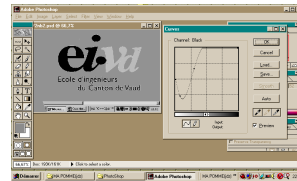




Robotique et automatisation, JDZ/JDD 20.04.2011

7

Exemple de traitement de pixel



eiv
Ecole d'ingénieurs
du Canton de Vaud



Robotique et automatisation, JDZ/JDD 20.04.2011

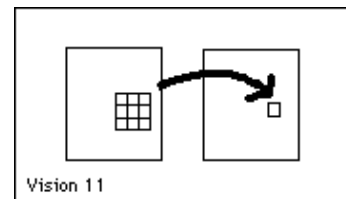
8

- **3 Traitement bidimensionnel d'images**
- 3.1 Opérations par pixels individuels (point par point) 28
- 3.2 Modifications des pixels d'après leur voisinage 29
- 3.3 Opérations globales ou par régions 39
- 3.4 Détection de mouvement 45
- 3.5 Transformée en cosinus 46
- 3.6 Temps de traitement 46

Robotique et automatisation, JDZ/JDD 20.04.2011

9

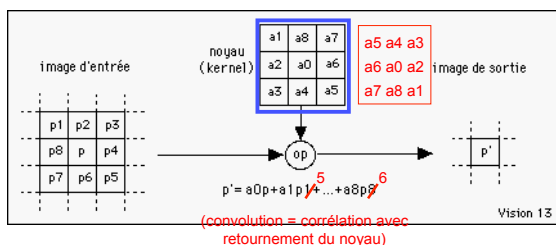
3.2 Modifications des pixels d'après leur voisinage



Robotique et automatisation, JDZ/JDD 20.04.2011

10

3.2.1 Filtres linéaires - principe

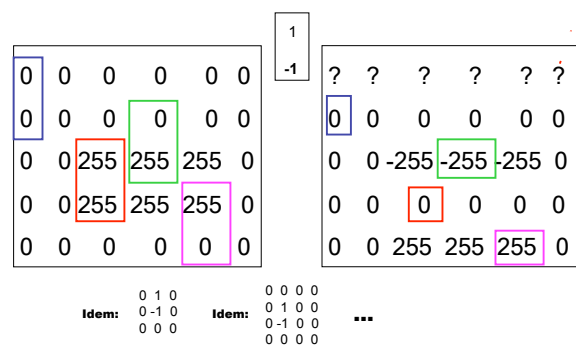


$$I'(l,c) = \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} k(i,j) * I(l-i, c-j)$$

Robotique et automatisation, JDZ/JDD 20.04.2011

11

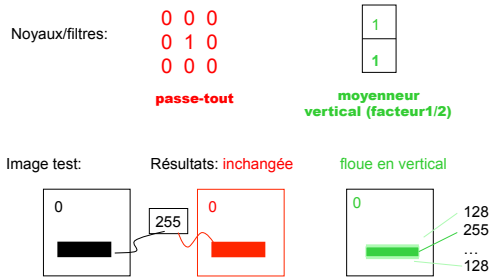
3.2.1 Filtres linéaires – exemple détaillé



Robotique et automatisation, JDZ/JDD 20.04.2011

12

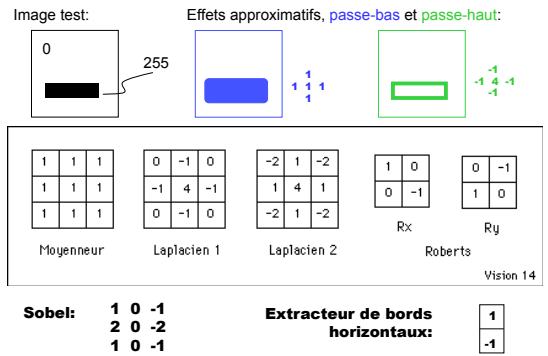
3.2.1 Filtres linéaires – cas de base



Robotique et automatisation, JDZ/JDD 20.04.2011

13

3.2.1 Filtres linéaires – cas de base



Robotique et automatisation, JDZ/JDD 20.04.2011

14

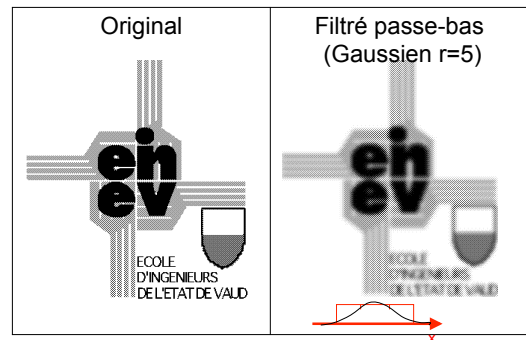
3.2.1 Filtres linéaires – Amplitudes ou poids du noyau



Robotique et automatisation, JDZ/JDD 20.04.2011

15

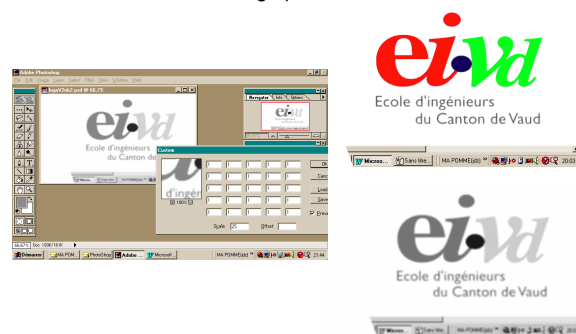
3.2.1 Filtres linéaires – exemples



Robotique et automatisation, JDZ/JDD 20.04.2011

16

3.2.1 Filtres linéaires – exemples Filtrage passe-bas



Robotique et automatisation, JDZ/JDD 20.04.2011

17

3.2.1 Filtres linéaires – exemples Extraction de bords verticaux

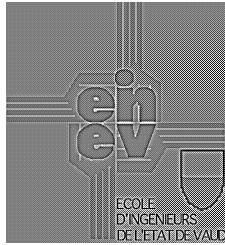


Robotique et automatisation, JDZ/JDD 20.04.2011

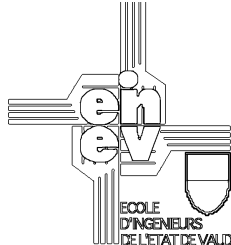
18

3.2.1 Filtres linéaires – exemples

Filtré passe-haut (r=4)



Extraction de bords



Robotique et automatisation, JDZ/JDD 20.04.2011

19

3.2.1 Filtres linéaires – exemples

Extraction de bords



Robotique et automatisation, JDZ/JDD 20.04.2011

20

3.2.1 Filtres linéaires – exemples

Extraction de bords



Robotique et automatisation, JDZ/JDD 20.04.2011

21

3.2.1 Filtres linéaires – composition de filtres élémentaires

$$[-1 \ 0 \ 1] = [-1 \ 1] * [1 \ 1]$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} ; \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Robotique et automatisation, JDZ/JDD 20.04.2011

22

3.2.1 Filtres linéaires – exemple: orientation des bords

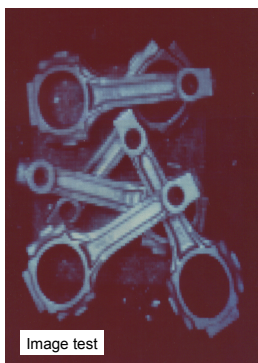
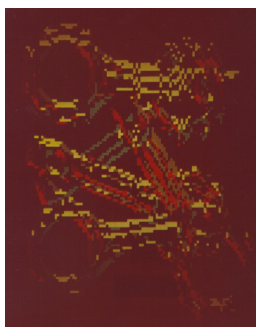


Image test



Résultats avec Roberts, orientation estimée sur tous les bords: $\text{Atan4}(R_x, R_y)$

Robotique et automatisation, JDZ/JDD 20.04.2011

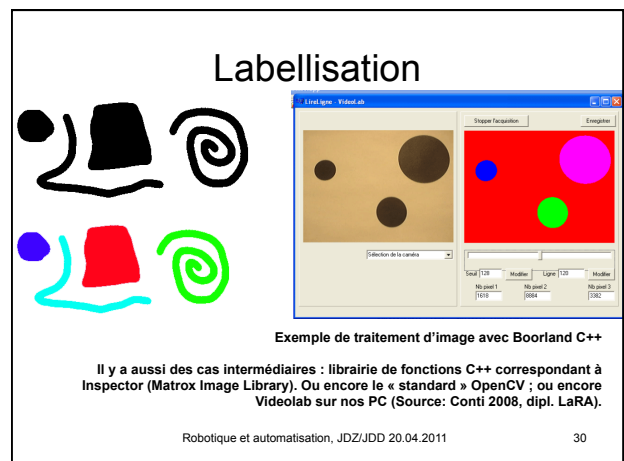
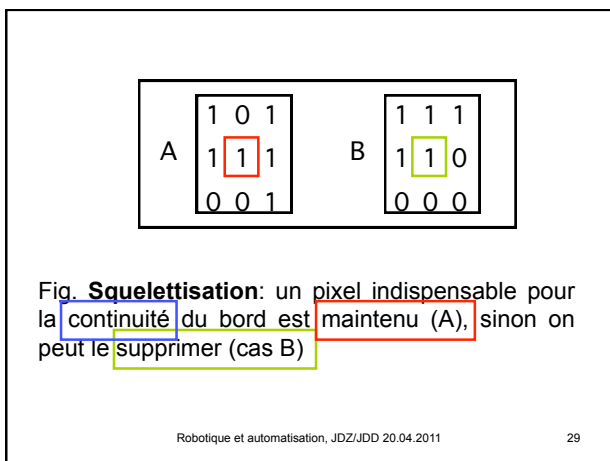
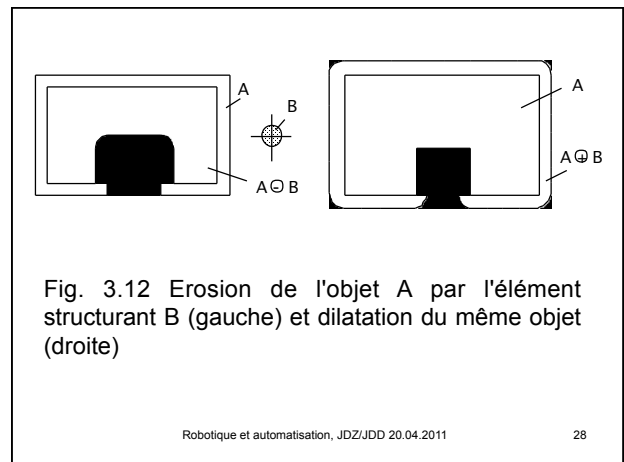
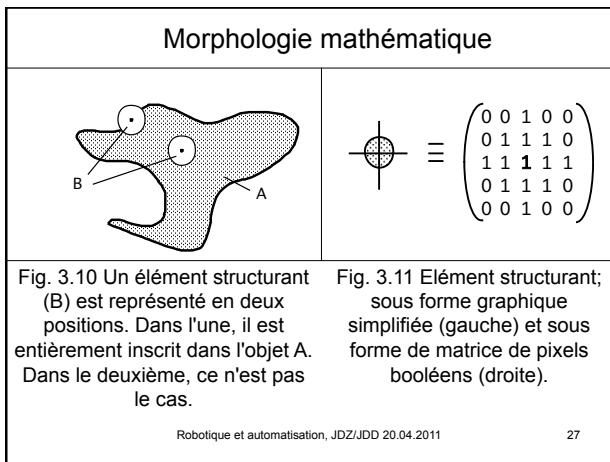
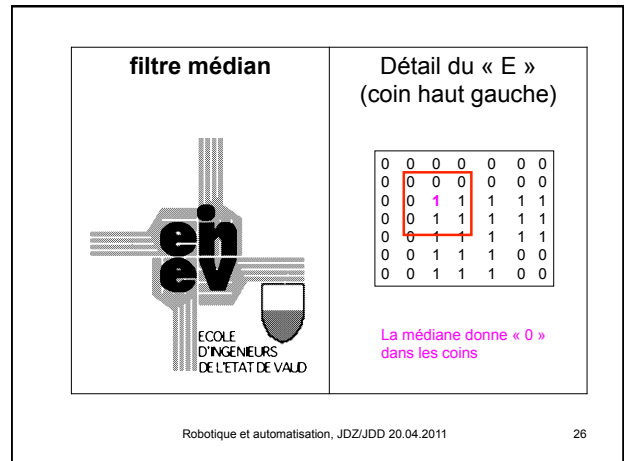
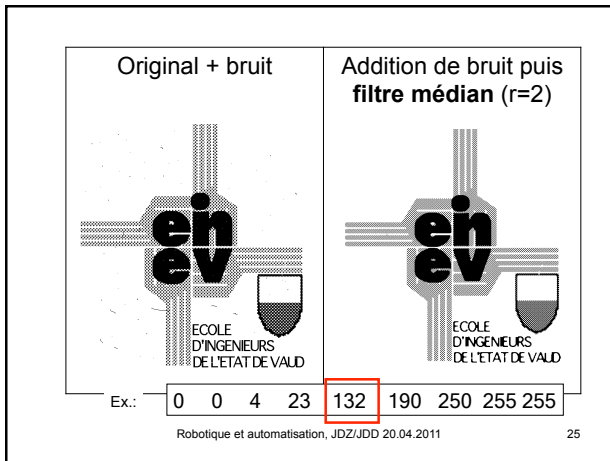
23

3.2.1 Filtres non-linéaires – cas de base

Original	Dilatation des 1 Érosion des 0																																																																								
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0																																																																				
0	0	0	0	0	0																																																																				
0	0	1	1	0	0																																																																				
0	0	0	0	0	0																																																																				
0	0	0	0	0	0																																																																				
0	0	0	0	0	0																																																																				
0	0	0	0	0	0																																																																				
0	0	1	1	0	0																																																																				
0	1	1	1	1	0																																																																				
0	0	1	1	0	0																																																																				
0	0	0	0	0	0																																																																				
0	0	0	0	0	0																																																																				

Robotique et automatisation, JDZ/JDD 20.04.2011

24



Labellisation

Principe de la méthode 1 - accès à 2 lignes à la fois

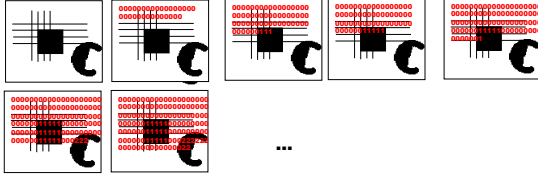
Traiter l'image de gauche à droite, de haut en bas:

-si blanc, avancer au pixel suivant

-si noir

- si pas de voisin noir, en haut ou à gauche: nouvelle étiquette (+1)

- si voisin noir, en haut ou à gauche: même étiquette



Robotique et automatisation, JDZ/JDD 20.04.2011

31

Labellisation

Principe de la méthode 2 - traitement récursif

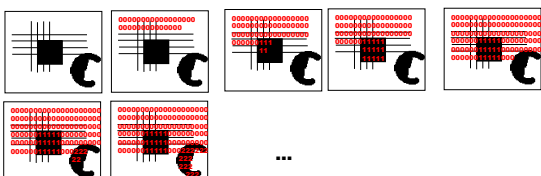
Traiter l'image de gauche à droite, de haut en bas:

-si blanc, avancer au pixel suivant

-si noir

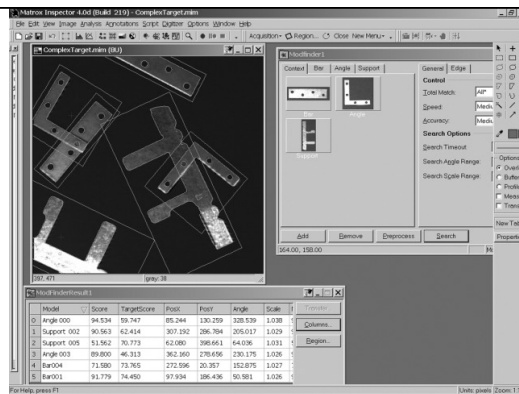
- nouvelle étiquette (+1)

- puis, de façon récursive, jusqu'à ce que tous les voisins noirs soient traités: même étiquette et même traitement pour ces voisins noirs (chaque pixel a 4 voisins immédiats).



Robotique et automatisation, JDZ/JDD 20.04.2011

32



Exemple de Matrox Inspector (source Conti 2008)

Robotique et automatisation, JDZ/JDD 20.04.2011

33

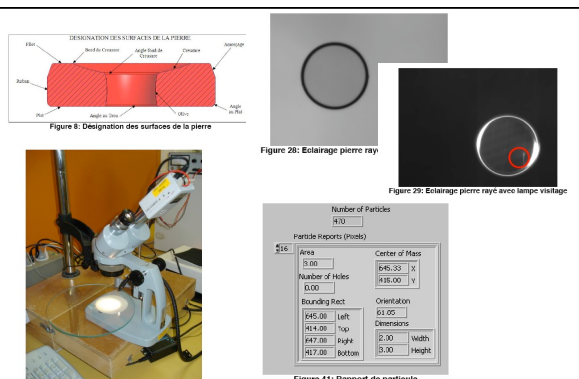


Figure 24: Implantation du test avec microscope

Exemple avec Labview (source Conti 2008)

Robotique et automatisation, JDZ/JDD 20.04.2011

34

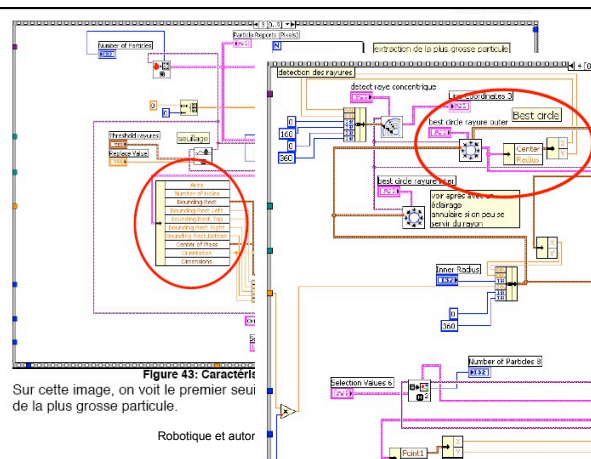
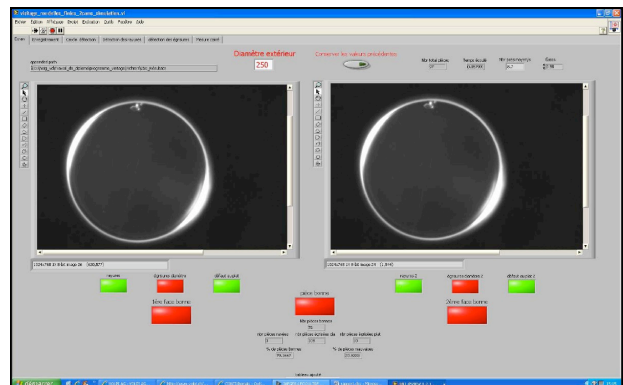


Figure 43: Caractéristiques de la plus grosse particule. Sur cette image, on voit le premier seuil de la plus grosse particule.

Robotique et automatisation, JDZ/JDD 20.04.2011

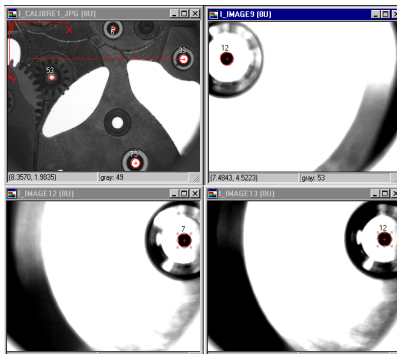


Exemple avec Labview (source Conti 2008)

Robotique et automatisation, JDZ/JDD 20.04.2011

36

Vision: Apprentissage de positions



N° du trou	Pos. X [mm]	Pos. Y [mm]
39	7.734	1.596
12 (haut)	0.776	1.551
7	7.769	1.646
12 (bas)	7.743	1.596

008)

Robotique et automatisé, JDZ/JDD 20.04.2011

37

Vision: Exemple Cognex pour prise de balle

	A	B	C	D	E	F	G	H	I
0	Image	acquisition de l'image							
1									
2	recherche 1 blob dans "Image", toute la surface est prise en compte								
3		Index	Ligne	Col	Angle	Couleur	Score	Surface	Elongation
4	Blobs	0.000	156.533	305.824	330.891	0.000	100.000	2490.000	0.000
5									
6	connexion programme externe, mode serveur								
7	Device								
8									
9	lecture message reçu								
10	trigger								
11	commande texte attendue pour le déclenchement								
12	trigger								
13	comparaison message								
14	1.000								
15	si pas d'erreur, déclenchement trigger manuel								
16	1.000	32.000							
17									
18	envoi message coordonnées blob								
19	Devote	156.5	305.8						

(source Manip. 39 Bosch-Delta, PFG, 2016)

Robotique et automatisé, JDZ/JDD 29.11.2016

38

3 Traitement bidimensionnel d'images

- 3.1 Opérations par pixels individuels (point par point)
- 3.2 Modifications des pixels d'après leur voisinage
- 3.3 Opérations globales ou par régions
- 3.4 Détection de mouvement
- 3.5 Transformée en cosinus
- 3.6 Temps de traitement

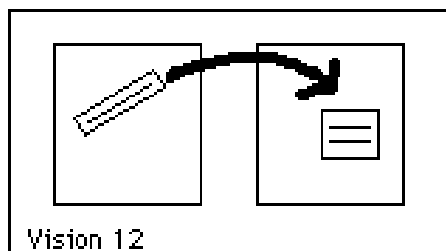
Robotique et automatisé, JDZ/JDD 20.04.2011

39

- 3.3 Opérations globales ou par régions
- Fenêtres avec déplacement, rééchantillonnage et rotation
- Transformation des coordonnées spatiales
- Codage (yc. transformées)
- Voir aussi Vision-Partie 3: reconnaissance de formes avec méthodes statistiques, telles que corrélation

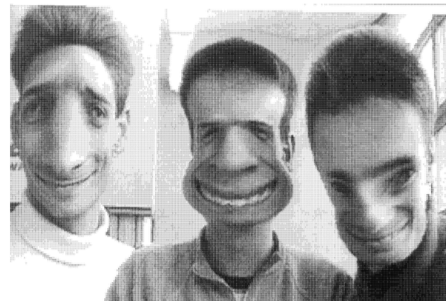
Robotique et automatisé, JDZ/JDD 20.04.2011

40



Robotique et automatisé, JDZ/JDD 20.04.2011

41

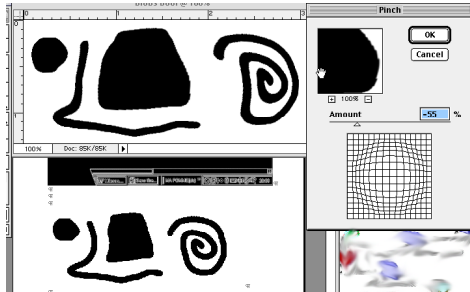


Traitement d'image avec opération sur les coordonnées spatiales ("Pinch")

Robotique et automatisé, JDZ/JDD 20.04.2011 / 29.01.2016

42

Traitement d'image avec opération sur les coordonnées spatiales - exemple



Robotique et automatisation, JDZ/JDD 20.04.2011

43

Code de plage - RLE

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

Code:

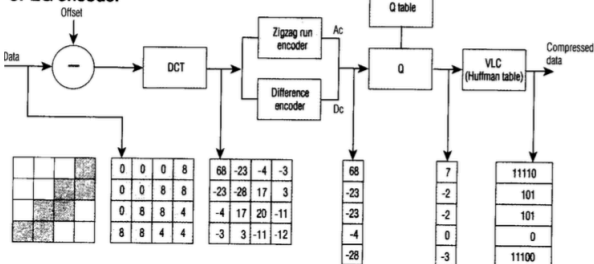
7
2,3,2
2,3,2
0,1,6
7

Note: Principe. Conventions: démarre par « 0 » ; codage par ligne.

Robotique et automatisation, JDZ/JDD 20.04.2011

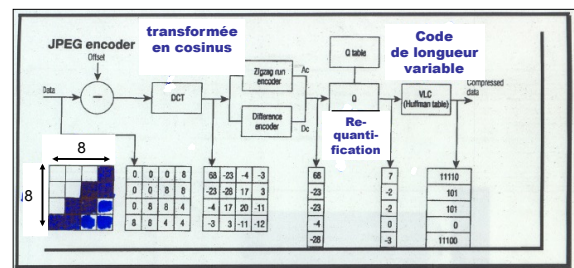
44

JPEG encoder



Robotique et automatisation, JDZ/JDD 20.04.2011 / 29.01.2016

45



Zigzag

Robotique et automatisation, JDZ/JDD 20.04.2011

46

Huffman table

Symbol	Code
0	0
-1	100
-2	101
+2	110
-3	11100
+1	11101
+7	11110

Robotique et automatisation, JDZ/JDD 20.04.2011 / 19.01.2016

47

Code de Huffman

- VLC (code de longueur variable)
- Code court pour éléments fréquents
- la longueur moyenne du code tend vers la limite théorique (minimum absolu)

ABABABABABACCD

1. Estimer les fréquences:
A B C D
7 6 2 1
2. Ordonner de façon croissante
1 2 6 7
3. Regrouper les deux valeurs les plus basses
1 et 2: 3 6 7
4. Réorganiser (ordonner à nouveau):
3 et 6: 9 7
7 9
7 et 9: 16

Robotique et automatisation, JDZ/JDD 20.04.2011

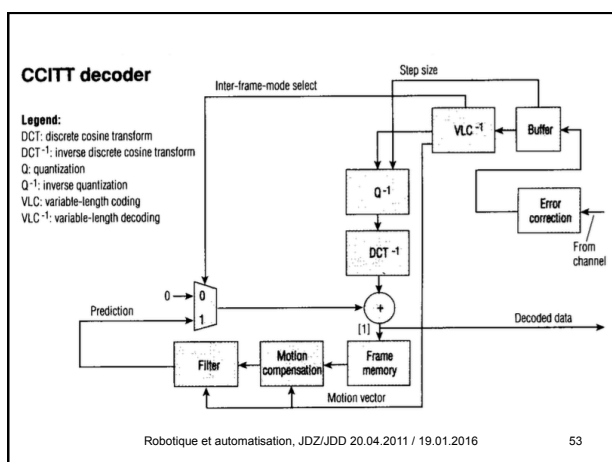
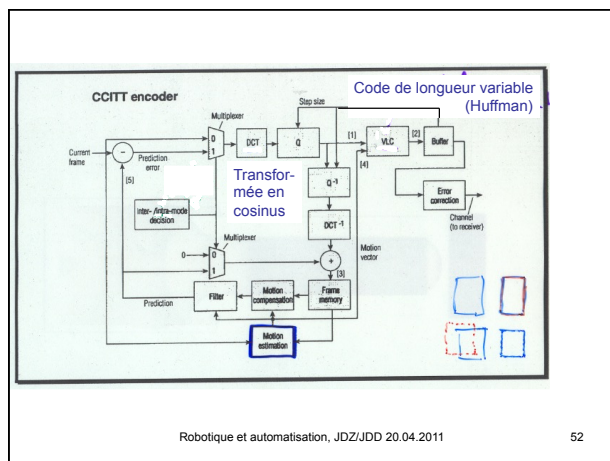
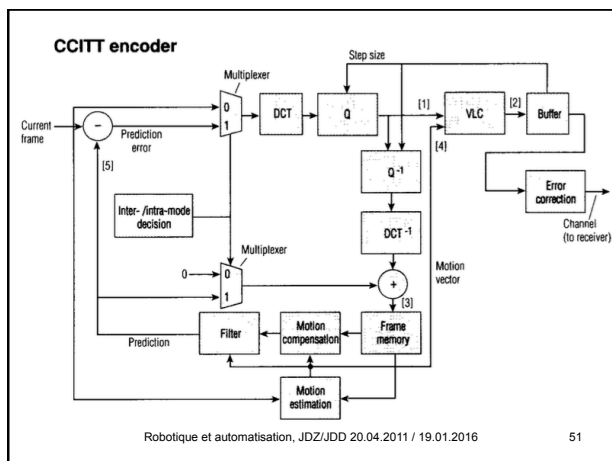
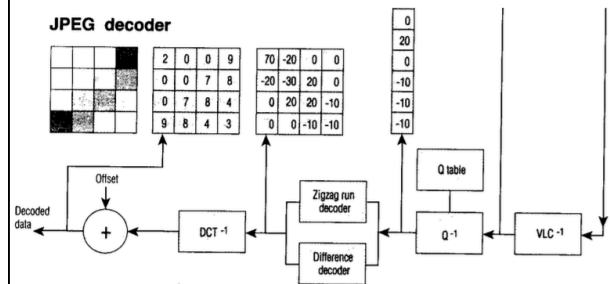
48


```

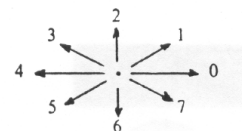
      16
    0/\1
      9 7 A
    0/\1
      3 6 B
0 0/\1
  1 2 C
  D

```

6. Conclusion:
A:1, B:01, C: 001, D: 000



Le code de Freeman est un code simple qui associe un nombre à chaque direction du plan selon le schéma suivant :



Dans ces conditions, un contour donné recevra une codification constituée par la suite des codes affectés à chaque direction élémentaire.

Par exemple, le contour :

sera codé : 0107700.

Evidemment, ne sont considérés que des angles multiples de $\pi/4$ puisqu'on travaille ici au niveau du pixel.

Ce type de code permettra, après traitement, de déterminer les contours des figures géométriques les plus variées.

Code de type LZW

CharStream input: THISISATESTTEST
Corresponding CodeStream: 19 7 8 18 28 0 19 4 18 19 32 34

Compress string table:

0	A	1	B	2	C	3	D	4	E
5	F	6	G	7	H	8	I	9	J
10	K	11	L	12	M	13	N	14	O
15	P	16	Q	17	R	18	S	19	T
20	U	21	V	22	W	23	X	24	Y
25	Z	26	TH	27	HI	28	IS	29	SI
30	ISA	31	AT	32	TE	33	ES	34	ST
35	TT	36	TES	37		38		39	

Reconstructed chaistream: T H I S I S A T E S T T E S T

Decompress string table:

0	A	1	B	2	C	3	D	4	E
5	F	6	G	7	H	8	I	9	J
10	K	11	L	12	M	13	N	14	O
15	P	16	Q	17	R	18	S	19	T
20	U	21	V	22	W	23	X	24	Y
25	Z	26	TH	27	HI	28	IS	29	SI
30	ISA	31	AT	32	TE	33	ES	34	ST
35	TT	36	TES	37		38		39	

Robotique et automatisaion, JDZ/JDD 20.04.2011

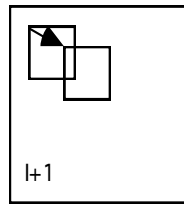
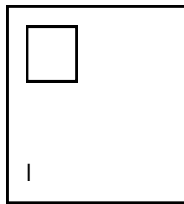
55

• 3 Traitement bidimensionnel d'images

- 3.1 Opérations par pixels individuels (point par point) 28
- 3.2 Modifications des pixels d'après leur voisinage 29
- 3.3 Opérations globales ou par régions 39
- 3.4 Détection de mouvement 45
- 3.5 Transformée en cosinus 46
- 3.6 Temps de traitement 46

Robotique et automatisaion, JDZ/JDD 20.04.2011

56



Robotique et automatisaion, JDZ/JDD 20.04.2011

57

• 3 Traitement bidimensionnel d'images

- 3.1 Opérations par pixels individuels (point par point) 28
- 3.2 Modifications des pixels d'après leur voisinage 29
- 3.3 Opérations globales ou par régions 39
- 3.4 Détection de mouvement 45
- 3.5 Transformée en cosinus 46
- 3.6 Temps de traitement 46

Robotique et automatisaion, JDZ/JDD 20.04.2011

58

3.5 Transformée en cosinus

La transformée en cosinus 2D d'une image $x(k,l)$ est donnée par l'équation suivante:

$$X(m,n) = \frac{4C(m)C(n)}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(k,l) \cos \frac{(2k+1)m\pi}{2N} \cos \frac{(2l+1)n\pi}{2N}$$

où $m, n = 0, \dots, N-1$, et les coefficients $C(m)$, $C(n)$ valent 1 (sauf si m ou $n = 0$, auquel cas ils valent $1/\sqrt{2}$).

La transformée inverse est donnée par l'équation suivante:

$$x(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} C(m)C(n)X(m,n) \cos \frac{(2k+1)m\pi}{2N} \cos \frac{(2l+1)n\pi}{2N}$$

Cette transformée s'opère sur des blocs 8x8 dans les normes JPEG et MPEG

Robotique et automatisaion, JDZ/JDD 20.04.2011

59

• 3 Traitement bidimensionnel d'images

- 3.1 Opérations par pixels individuels (point par point) 28
- 3.2 Modifications des pixels d'après leur voisinage 29
- 3.3 Opérations globales ou par régions 39
- 3.4 Détection de mouvement 45
- 3.5 Transformée en cosinus 46
- 3.6 Temps de traitement 46

Robotique et automatisaion, JDZ/JDD 20.04.2011

60

- **3 Traitement bidimensionnel d'images**
- 3.1 Opérations par pixels individuels (point par point)
- 3.2 Modifications des pixels d'après leur voisinage
- 3.3 Opérations globales ou par régions
- 3.4 Détection de mouvement
- 3.5 Transformée en cosinus
- 3.6 Temps de traitement
- 3.7 Conclusion concernant le traitement 2D

3.7 Conclusion concernant le traitement 2D

- Traitement simple et « parfait » au niveau des pixels indépendamment les uns des autres
- Traitement très courant et classique en théorie et dans les produits du marché (logiciel, matériel), lorsque des noyaux sont considérés
 - Limites:
 - résolution faible si le noyau est grand;
 - variété de traitements faible si le noyau est petit
- Cas spéciaux: labellisation, codage, travail dans des fenêtres à résolution x-y paramétrable, réorientées