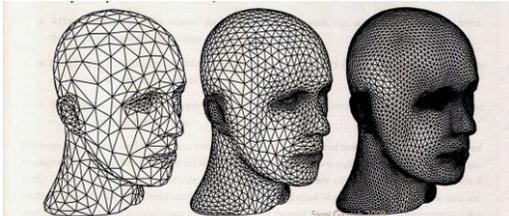




# INFOGRAPHIE & TRAITEMENT NUMÉRIQUE D'IMAGES

Dr. Sami GAZZAH  
ISITCOM Hammam Sousse



Sami Gazzah 2009



## Plan du cours

**Objectifs:**

- Maîtriser les concepts de base liés à l'imagerie numérique 2D sur ordinateur.
- Développer une méthodologie de traitement numérique d'images et de manipulation d'objets graphiques.
- Comprendre et appliquer les notions d'infographie 2D et 3D dans le cadre du développement et de l'utilisation d'un logiciel graphique;



Sami Gazzah 2009

## Contenu du cours

1. Introduction : traitement d'image Versus Infographie
2. Transformations sur les images
3. Restauration d'Images
4. Rehaussement d'images
5. Segmentation d'images
6. Introduction aux opérateurs morphomathématique
7. Transformations géométriques 2D
8. Génération de segments de droite et de cercles en 2D
9. introduction au traitement d'images avec OpenGL
10. introduction au traitement d'images avec OpenCV

Sami Gazzah 2009

## Travaux pratiques

Six travaux pratiques pour apprendre à manipuler et traiter des images. Les travaux sont effectués en langage MATLAB.

- ⊕ TP1 : Manipulation d'images niveaux de gris et couleurs
- ⊕ TP2: Manipulation sur l'analyse élémentaire d'image (Hist.)
- ⊕ TP2 : Compression
- ⊕ TP3 : Restauration d'images par transformations géométriques
- ⊕ TP6 : Filtrage : détection de contours, lissage, etc

Sami Gazzah 2009



## Chapitre I :

### Traitement d'image Versus Infographie



Sami Gazzah 2009

## I. Traitement d'image Vs Infographie

### I.1 Traitement d'image

Pour l'être humain, l'image constitue l'une des principales **sources d'informations**.

Une scène contient une **quantité importante d'informations** qu'il est utile parfois de **traiter** et **d'analyser** dans le but de prendre des **décisions**.



Sami Gazzah 2009

## I. Traitement d'image Vs Infographie

### I.2 Historique.....

#### 1950-1970 : Traitement de l'image :

L'existence d'images de mauvaises qualités (Rayons X, OCR, ...) et volumineuses a conduit les chercheurs à s'intéresser à :

- La **restauration** (corriger les défauts liés à l'acquisition),
- L'**amélioration** (rendre l'image "belle" pour l'affichage),
- La **compression** (réduire le volume de l'image)

Sami Gazzah 2009

7

## I. Traitement d'image Vs Infographie

### I.2 Historique.....

#### 1970-1980 : Du traitement vers l'interprétation :

L'évolution naturelle vers l'extraction automatique d'informations a conduit à l'apparition de nouvelles notions telles que :

- L'analyse d'images en vue de l'extraction automatique d'information d'où les thèmes : seuillage, segmentation, extraction de contours, morphologie mathématique (pour les images binaires), ...
- L'interprétation d'images et les systèmes experts

Sami Gazzah 2009

8

## I. Traitement d'image Vs Infographie

### I.2 Historique.....

#### 1980- : La vision par ordinateur

- De l'image 2D aux modèles tri- dimensionnels
- Analyse du mouvement
- Vision pour la robotique (mouvement, 3D, détection d'obstacle, trajectoire)
- les bibliothèques numériques
- Transmission (codage et réseaux)

Sami Gazzah 2009

9

## les bibliothèques numériques : Reconnaissance de l'écriture OCR

Caractère imprimé	Caractères écrits par 10 scripteurs différents
م	
ي	
ك	



- Traitement automatique des chèques
- Tri- postale

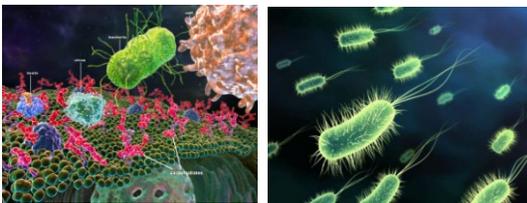
Sami Gazzah 2009

10

## Imagerie biologique

Imagerie biologique:

- identification des cellules, bactéries, virus
- comptage des cellules



Sami Gazzah 2009

11

## Imagerie médicale

Imagerie médicale:

- Aide au diagnostique
- Reconstruction 3D des images

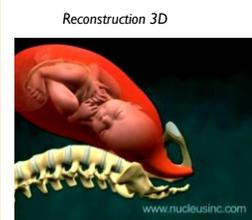
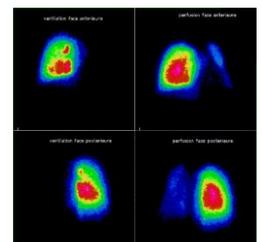


Image de scintigraphie cardiaque



Sami Gazzah 2009

12

f6cilusq

## Vision robotique

Vision des robots :

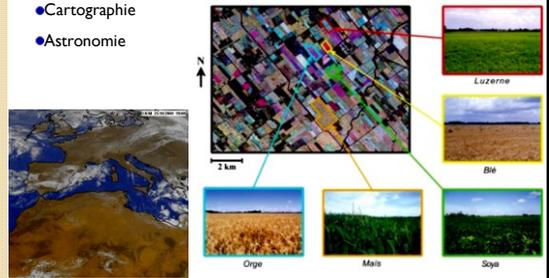
- Robots autonomes
  - Intervention en milieu contaminé
  - Sonde d'exploration
- Robotique médical
  - Assistance au geste médical



Sami Gazzah 2009 13

## Imagerie aérienne et satellitaire

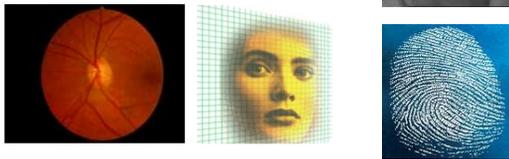
- Analyse des ressources terrestres
- Météo
- Cartographie
- Astronomie



Sami Gazzah 2009 14

## Biométrie

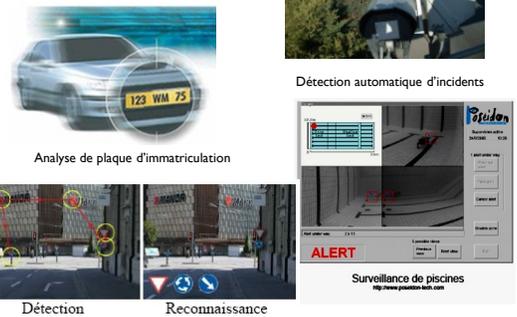
- Reconnaissance d'empreintes digitales
- Reconnaissance de visage
- Reconnaissance d'iris
- Reconnaissance du style d'écriture du scripteur



Sami Gazzah 2009 15

## Vidéosurveillance

- Détection de mouvements et reconnaissance de comportement



Détection Reconnaissance Sami Gazzah 2009 16

## Infographie

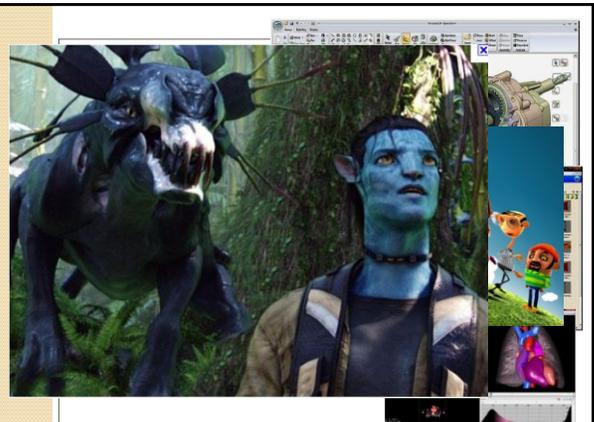
Informatique & Graphisme

Ensemble de techniques permettant de créer et de manipuler des images synthétiques.

Produit final → image synthétique.



Sami Gazzah 2009 17



Sami Gazzah 2009 18

## I. Traitement d'image Vs Infographie

Sami Gazzah 2009 19

## 2.Acquisition des images (1)

Parmi les capteurs on peut distinguer:

- les capteurs thermiques (thermopiles)
- les films photographiques
- les capteurs photoélectriques (photodiodes, CCD, CMOS,...)

Le signal obtenu est caractérisé par:

- sa dimension : 1D, 2D ou 3D
- sa nature

Sami Gazzah 2009 20

## 2.Acquisition des images (2)

- La représentation informatique d'une image est nécessairement **discrète** alors qu'une image est de nature **continue**.
- Certains capteurs effectuent une discrétisation.
- La transformation d'un signal analogique 2D nécessite :
  - Une discrétisation de l'espace: c'est l'**échantillonnage**.
  - Une discrétisation des couleurs: c'est la **quantification**.

Sami Gazzah 2009 21

## 2.Acquisition des images (3)

1.  $I(x,y)$  continue

2.  $I(x,y)$  numérisée (échantillonnée et quantifiée)

Scène Image Numériseur (CCD, CMOS) Image numérique

1. Projection 2D d'une scène 3D

2.  $I(x,y)$  représente l'intensité de la lumière au point  $(x,y)$

3. Discrétisation de l'espace et de l'intensité

0	15	10	15	30	70	80
0	0	100	120	120	100	120
0	20	160	160	160	80	80
0	15	70	100	10	20	20
0	15	70	0	0	0	10
0	15	50	120	110	100	110
0	10	20	30	30	20	20

Sami Gazzah 2009 22

## 2.Acquisition des images (4)

Sami Gazzah 2009 23

## 2.Acquisition des images (5)

Signal lumineux → Capteur CCD → Signal électrique

**CCD (Charge-Coupled Device)**

- Senseur de haute qualité: bruit faible, sensibilité à la lumière, haute résolution
- Forte consommation d'énergie
- Technologie coûteuse
- Il faut transférer toute la matrice de pixels avant de pouvoir y accéder

Sami Gazzah 2009 24

### 3. Représentation d'une image numérique

Pour numériser des images, deux opérations :

- ☐ échantillonnage
- ☐ quantification

x =	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	
y =	41	210	209	204	202	197	247	143	71	64	80	84	54	64	57	58
	42	200	196	203	197	195	210	207	56	63	58	53	61	62	51	
	43	201	207	182	201	198	213	198	69	65	57	52	53	50	50	
	44	216	205	211	193	202	207	200	57	69	60	55	77	48	61	
	45	221	206	211	194	199	197	220	56	63	60	55	49	67	56	
	46	209	214	224	159	184	153	204	173	64	60	59	51	62	48	
	47	204	212	213	208	191	190	191	214	60	62	66	76	51	49	
	48	214	215	215	207	208	180	172	188	69	72	55	49	56	56	
	49	209	205	214	205	204	195	187	185	86	82	68	87	82	48	
	50	208	209	203	202	198	174	185	149	71	63	55	55	45	46	
	51	207	210	211	159	217	104	193	172	209	90	62	84	62	52	
	52	206	205	209	209	197	194	193	187	167	220	59	61	51	50	
	53	204	206	203	209	195	203	188	185	183	221	76	61	58	60	
	54	200	203	193	205	189	197	193	192	163	195	122	63	64	65	
	55	205	210	202	203	199	197	190	181	173	180	105	62	64	63	

Sami Gazzah 2009 25

### 3. Représentation d'une image numérique

L'échantillonnage est le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire  $R(x,y)$  d'une image continue une unique valeur  $I(x,y)$ .

On parle de *sous-échantillonnage* lorsque l'image est déjà discrétisée et qu'on diminue le nombre d'échantillons.

La *quantification* désigne la limitation du nombre de valeurs différentes que peut prendre  $I(x,y)$ .

Une *image numérique* est une image échantillonnée et quantifiée.

Sami Gazzah 2009 26

### 3. Représentation d'une image numérique

Lena et un détail de l'image

112	77	56	66	157	204	204	198
108	74	53	87	177	207	204	204
72	57	66	126	197	209	206	208
55	65	113	173	207	211	210	211
77	107	160	198	208	206	208	207
127	158	188	202	203	202	202	206
163	186	197	198	202	205	207	209
182	187	193	197	199	203	206	209

Valeurs du détail de l'image Lena

Sami Gazzah 2009 27

### 3. Représentation d'une image numérique

Discretisation de l'espace 2D, on découpe l'image en pixels

problèmes d'aliasing !!

pixel (picture element)

(1) Maillage (2) Quantification

Sami Gazzah 2009 28

### 3. Représentation d'une image numérique

Une résolution trop faible peut causer des problèmes d'aliasing !!

→ Apparition d'escaliers sur les contours obliques

512x512 256x256 128x128  
64x64 32x32 16x16

Sami Gazzah 2009 29

### 3. Représentation d'une image numérique

Le pixel  $[i,j]$

$I[i,j] = N$

Valeur Niveau de gris

$N \in [N_{min}, N_{max}]$

$(N_{max} - N_{min}) = \text{nombre de niveaux de gris}$

Une image numérique  $I$

Sami Gazzah 2009 30

### 3. Représentation d'une image numérique

Discrétisation de l'espace des couleurs ou niveaux de gris  
 ⇒ Une quantification trop faible peut causer des problèmes de **faux contours**

255	80	255	255	255
255	80	255	255	255
255	80	70	60	255
255	50	255	40	255
255	40	20	0	255

Image en niveaux de gris et sa représentation matricielle

Sami Gazzah 2009 31

### 3. Représentation d'une image numérique

#### 3.1. Pavage ou tessellation

A chaque échantillonneur (sommet du maillage), on associe une surface élémentaire dite "pavé" ou "tesselle".

Les pixels sont généralement arrangés sous forme rectangulaire, dans un tableau 2D ... il peut exister d'autres pavages ou tessellations :

- ❑ Le pavage hexagonale,
- ❑ Le pavage triangulaire ou carré
- ❑ Le pavage rectangulaire en quinconce.

Pavage triangulaire  
 Pavage en quinconce  
 Pavage carré  
 Pavage hexagonale

Sami Gazzah 2009 32

### 3. Représentation d'une image numérique

#### 3.1. Pavage ou tessellation... choix...motivation

Certains pavage sont motivés par :

- ✓ leurs adéquation avec le système d'acquisition: exemple le pavage quinconce correspond à certains systèmes d'acquisitions embarqués dans les satellites (satellite spot 5).
- ✓ des relations géométriques telles que le voisinage: nombre des pixels voisins connexes. (4, 6, 8, 32,...).

4 voisins  
 6 voisins  
 8 voisins

Sami Gazzah 2009 33

### systèmes d'acquisitions du satellite spot 5

Introduction du **Super-Mode**: permet la création d'une image à 2,5 mètres de résolution à partir de deux images à 5 mètres (acquises simultanément avec un demi-pixel de décalage)

5-METRE SAMPLING GRID FOR EACH IMAGE  
 2.5-METRE SAMPLING GRID

Image 1  
 Image 2  
 INTERLACING  
 INTERPOLATION  
 RESTORATION

Orange grid: pixels acquired by first line detectors.  
 Blue grid: pixels acquired by second line detectors.

Source : [http://www.eoportal.org/directory/pres\\_SPOT5.html](http://www.eoportal.org/directory/pres_SPOT5.html) 29 OCT 2010 34

### 3. Représentation d'une image numérique

#### 3.2. Codage des pixels

- ❑ Pour un ordinateur une image ⇒ ensemble de pixels
- ❑ Un pixel possède une valeur : scalaire, vecteur, ...
- ❑ Ordre de grandeur :
  - si un pixel est codé sur 8 bits ⇒  $2^8=256$  couleurs ou niveau de gris
  - si .....sur 16 bits ⇒  $2^{16}=65\ 536$  couleurs...plus d'espace
  - .....24 bits ⇒  $2^{24}=16\ 777\ 216$  couleurs...encore ++

Sami Gazzah 2009 35

### 3. Représentation d'une image numérique

#### 3.3. Distance entre pixels

Différents distances entre 2 points  $P(i,j)$  et  $P(k,l)$  peuvent être utilisées:

1. Distance euclidienne :

$$\delta(P(i,j), P(k,l)) = \sqrt{(k-i)^2 + (l-j)^2}$$

+ Cette distance ne privilège aucun axe

Sami Gazzah 2009 36

### 3. Représentation d'une image numérique

#### 3.3. Distance entre pixels

Différents distances entre 2 points  $P(i,j)$  et  $P(k,l)$  peuvent être utilisées:

2. Distance blocks:  
Cette distance est appelé également « Manhattan distance »  
il faut contourner les blocs d'habitations pour relier deux points

$$dist(P(i,j), P(k,l)) = |k - i| + |l - j|$$

Sami Gazzah 2009 37

### 3. Représentation d'une image numérique

#### 3.3. Distance entre pixels

Différents distances entre 2 points  $P(i,j)$  et  $P(k,l)$  peuvent être utilisées:

3. Distance tour d'échiquier:

$$dist(P(i,j), P(k,l)) = \max(|k - i|, |l - j|)$$

Sami Gazzah 2009 38

— distance euclidienne  
— distance blocs  
— distance tour d'échiquier

Sami Gazzah 2009 39

### 4. Formats d'images

#### 4.1. Image vectorielle Vs. Image matricielle

- Une image vectorielle est représentée par des formes géométriques simples :
  - ✓ Ligne : point de départ, point d'arrivée, ...
  - ✓ Triangle : 3 sommets, la couleur du contour, la couleur intérieure (remplissage), ...
 Ex: PAO (publication assistée par ordinateur), DAO (dessin assisté par ordinateur), etc
- Une image matricielle est représentée par des matrices de pixels Bitmap (image en mode point)
  - Ex: image en niveaux de gris ou couleurs de type photo : GIF, PCX, BMP, JPEG, PPM, PGM...

Sami Gazzah 2009 40

### 4. Formats d'images

#### 4.1. Images vectorielles Vs. Images matricielles

Images vectorielles		Images matricielles	
Avantages	Inconvénients	Avantages	Inconvénients
Peu de place en mémoire.	Peut devenir gros pour une image complexe.	Aucun calcul pour l'affichage.	Peuvent occuper beaucoup d'espace mémoire.
Redimensionnement sans perte d'informations.	Beaucoup de calculs pour l'affichage.	La complexité de l'image n'influe pas sur sa taille.	
Aucun aliasing si on manipule l'objet graphique.			

Sami Gazzah 2009 41

### 4. Formats d'images

#### 4.2. Stockage des images

- Pour la mémorisation des images sur disque mémoire : on stocke les informations concernant les image dans un en-tête puis les données (valeurs de pixels) .
- L'en-tête contient la largeur et la hauteur de l'image aussi on peut ajouter d'autres informations telles que le type de données, l'auteur, la date, ...
- Pour certains formats l'en-tête et les données sont stockés dans 2 fichiers différents.
- Les données sont souvent stockées dans l'ordre des pixels de gauche à droite et de haut en bas.
- Les anciens formats sont simples et très facile à lire et à écrire :
  - PBM : pour des images binaires
  - PGM : pour des images en niveaux de gris
  - PPM : pour des images couleurs
- Les formats les plus courants sont : GIF, PCX, BMP, TIFF et JPEG.

Sami Gazzah 2009 42



### Images couleurs (24 bits)

#### L'espace couleur Rouge Vert Bleu (RGB)

Sami Gazzah 2009 49

Valeur R	Valeur V	Valeur B	Couleur correspondante	Commentaires
0	0	0		noir
...	...	...	...	...
0	0	255		bleu
...	...	...	...	...
0	255	0		vert
...	...	...	...	...
255	0	0		rouge
...	...	...	...	...
128	128	128		couleur intermédiaire correspondant à un gris
255	255	255		blanc

Sami Gazzah 2009 50

Sami Gazzah ISITCom 2009 51

#### L'espace couleur Cyan, Magenta et Jaune (GMJ)

- les couleurs primaires CMJ sont ici les complémentaires des RVB
- utilisé par certaines imprimantes couleurs à jet d'encre
- les différentes couleurs sont obtenues par **soustraction du blanc**

$$\begin{pmatrix} C \\ M \\ J \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ V \\ B \end{pmatrix}$$

Sami Gazzah 2009 52

#### Images HSV

Le principe de l'espace HSV est de caractériser les couleurs de façon plus intuitive, conformément à la perception naturelle des couleurs, en termes de :

- teinte (Hue):** intuitivement, c'est le nom qu'on utilisera pour désigner la couleur, "vert", "mauve", etc. Idéalement associé à une longueur d'onde,
- saturation (Saturation):** c'est le taux de pureté de la couleur, qui doit varier entre la pureté maximale (couleur éclatante) et l'achromatisme (niveau de gris).
- valeur (Value):** c'est la mesure de l'intensité lumineuse de la couleur, qui doit varier entre le noir absolu et le blanc.

Sami Gazzah 2009 53

Pour passer de la représentation initiale (RGB) à cette représentation, on opère de la manière suivante:

$$A = U_1 \cdot (a \log(R) + b \log(V) + c \log(B))$$

$$C_1 = U_2 \cdot (\log(R) - \log(V))$$

$$C_2 = U_3 \cdot (\log(B) - 0.5 \log(R) - 0.5 \log(V))$$

U<sub>1</sub>, U<sub>2</sub> et U<sub>3</sub> sont des facteurs de normalisation.  
 Les valeurs des coefficients (a,b,c) dépendent des auteurs. Les valeurs courantes sont : (a,b,c) = (0.612, 0.369, 0.019)

Puis passage de AC<sub>1</sub>C<sub>2</sub> à HLS en appliquant les formules suivantes :

$$L = A$$

$$S = (C_1^2 + C_2^2)^{1/2}$$

$$H = \arccos(C_1/S)$$

Sami Gazzah 2009 54



# Chapitre 2 :

## Transformations sur les images

Sami Gazzah 2009 61

### Introduction

Au cours du processus de traitement et d'analyse, l'image subit une série de transformations.

Les transformations peuvent être classées de la manière suivante:

- Transformations ponctuelles.
- Transformations de voisinage.
- Transformations spectrales.
- Transformations morphologiques.

Sami Gazzah 2009 62

### I. Transformations ponctuelles

A partir d'une image source, la nouvelle valeur du pixel dépend uniquement de son ancienne valeur.  $I_1(x,y), I_2(x,y) \rightarrow I'(x,y) = f(I_1(x,y), I_2(x,y))$

**Exemple :** seuillage, ajustement luminosité/contraste, opérations algébriques, opérations logiques et arithmétiques, manip. d'histogramme.

**I. Opérations logiques et arithmétiques**

**Les images opérands**

- doivent être de même taille ;
- peuvent être des images constantes.

**Exemples**

- Addition, soustraction, ...
- ET logique, OU logique, ...

**Problèmes**

- débordements de [0, 255]
- normalisation

Sami Gazzah 2009 63

□ Images binaires : ET/OU logique entre les pixels.

$I_1(x,y)$	$I_2(x,y)$	$ET(I_1, I_2)(x,y)$	$OU(I_1, I_2)(x,y)$	$XOR(I_1, I_2)(x,y)$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

□ Images en niveaux de gris : ET/OU logique bit à bit entre les 8 bits codant les valeurs des pixels.

bit n°	7	6	5	4	3	2	1	0
$I_1(x,y) = 73$	0	1	0	0	1	0	0	1
$I_2(x,y) = 141$	1	0	0	0	1	1	0	1
$ET(I_1, I_2)(x,y) = 9$	0	0	0	0	1	0	0	1
$OU(I_1, I_2)(x,y) = 205$	1	1	0	0	1	1	0	1
$XOR(I_1, I_2)(x,y) = 196$	1	1	0	0	0	1	0	0

Sami Gazzah 2009 64

☛ **Les opérations logiques**

- ET:  $I(x,y) = I_1(x,y) \text{ ET } I_2(x,y)$  ➔ intersection logique
- OU:  $I(x,y) = I_1(x,y) \text{ OU } I_2(x,y)$  ➔ union logique
- NOT :  $I(x,y) = NOT(I_1(x,y))$

☛ **opérations arithmétiques**

- Addition:  $I(x,y) = I_1(x,y) + I_2(x,y)$ ;
- Soustraction:  $I(x,y) = I_1(x,y) - I_2(x,y)$ ;
- Multiplication,
- Division par une constante...

Sami Gazzah 2009 65

### Les opérations logiques

complémentaire de A = NON logique

A ∩ B = ET logique

A ∪ B = OU logique

A ⊕ B = OU exclusif

A - B = différence logique

Sami Gazzah 2009 66

**Application : Masquage**

$I_1$        $I_2$  (masque)

ET ( $I_1, I_2$ )      OU ( $I_1, I_2$ )

Sami Gazzah 2009 67

**Les Opérations arithmétiques**

**Addition d'images**

**Principe :**  $I'(x,y) = I_1(x,y) + I_2(x,y)$  pour tout pixel de coordonnées  $(x,y)$

**Stratégies si dépassement de capacité**

- Décalage des valeurs dans  $[0, 127]$  avant addition (perte du bit de poids faible)
- Saturation :  $I'(x,y) = \min(I_1(x,y) + I_2(x,y), 255)$
- Pré-calcul des valeurs finales (théoriques) minimale et maximale :
 
$$i'_{\min} = \min_{(x,y) \in I} (I_1(x,y) + I_2(x,y))$$

$$i'_{\max} = \max_{(x,y) \in I} (I_1(x,y) + I_2(x,y))$$
 puis recadrage de la dynamique :
 
$$I'(x,y) = \frac{255}{i'_{\max} - i'_{\min}} (I_1(x,y) + I_2(x,y))$$

**Utilisations principales**

Augmentation de la luminance d'une image (par addition d'une constante ou d'une image avec elle-même) et diminution du bruit dans une série d'images

Sami Gazzah 2009 68

**Les Opérations arithmétiques**

**Soustraction d'images**

**Principe :**  $I'(x,y) = I_1(x,y) - I_2(x,y)$  pour tout pixel de coordonnées  $(x,y)$

**Stratégies si dépassement de capacité**

- Saturation :  $I'(x,y) = \max(I_1(x,y) - I_2(x,y), 0)$
- Différence absolue :  $I'(x,y) = |I_1(x,y) - I_2(x,y)|$

**Utilisations principales**

- Diminution de la luminance d'une image
- Détection de changements entre images
  - défauts (par comparaison avec une image de référence)
  - mouvements (par comparaison avec une autre image de la séquence)

Sami Gazzah 2009 69

Entrance      Interspace      Kitchen

Rim Romdhaneet al., AVSS'10, 4<sup>th</sup> the 7th IEEE International Conference on Advanced Video and Signal-Based Surveillance, Boston, USA, 2010

Sami Gazzah 2009 70

(a) Image originale  $I_1$       (b) Image originale  $I_2$

Addition et soustraction de deux images.

Sami Gazzah 2009 71

Si on dispose de l'image en niveau de gris et de l'image binaire suivante

a. Image Niveau de gris      b. Image binaire

c. Image résultante

Sami Gazzah 2009 72

2. Identité, négatif et Seuilage

**Exemples**

**identité**  $i' = i$

**négatif**  $i' = 255 - i$

**seuilage**  $i' = \begin{cases} 0 & \text{si } i < 127 \\ 255 & \text{si } i \geq 127 \end{cases}$

$i$	$i' = f(i)$
0	0
1	1
2	2
...	...

$i$	$i' = f(i)$
0	255
1	254
2	253
...	...

$i$	$i' = f(i)$
0	0
1	0
2	0
...	...

2. Les opérations géométriques

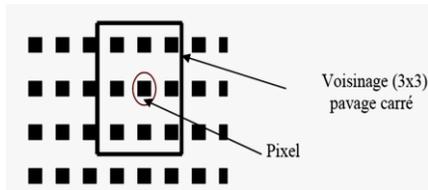


Opérations géométriques sur une image en niveaux de gris

3. Transformations de voisinage

Une transformation de voisinage est telle que la nouvelle valeur d'un pixel tient compte des pixels appartenant au **voisinage** du pixel considéré.

Le voisinage peut avoir une taille plus ou moins grande (3x3, 5x5, 7x7...), et il peut avoir une forme régulière (carré, rectangle, hexagone...) ou bien une forme quelconque



Exemple de transformation de voisinage : le produit de convolution

dans le cas continu, le produit de convolution de  $f$  par  $g$  est défini par :

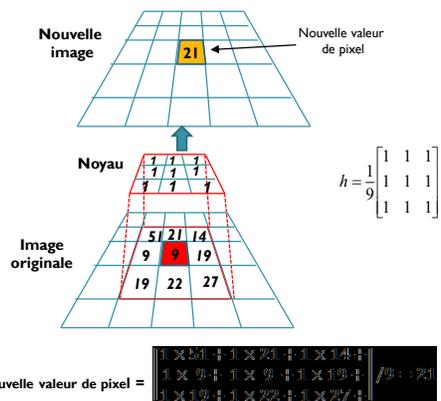
$$f * g(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) g(u - x, v - y) du dv$$

Dans le cas discret, il est généralement défini sur un voisinage, il est donné par :

$$f * g(x, y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(x, y) g(u - x, v - y)$$

Si on applique ceci à une image  $I_1$  de dimensions finies et à moyenne de convolution  $K$  ( $K$  : Kernel=noyau) de dimension  $3 \times 3$  les pixels de l'image  $I_1$  obtenue par convolution de  $I_2$  par  $K$  ont pour valeur :

$$I_2(i, j) = \sum_{k=0}^2 \sum_{l=0}^2 I_1(i + 1 - k, j + 1 - l) K(k, l)$$



## 4- Transformations spectrales (1)

Transformée de Fourier 2D (T.F.2D)

### Dans la cas 1D (Rappel)

- La T.F. est :  $F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx$
- La T.F inverse :  $f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux} du$

### Dans la cas 2D

- La T.F. est :  $F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)e^{-j2\pi ux} e^{-j2\pi vy} dx dy$
- La T.F inverse :  $f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u,v)e^{j2\pi ux} e^{j2\pi vy} du dv$

Sami Gazzah 2009

79

## 4- Transformations spectrales (2)

Exemple des transformations spectrales : **transformée de Fourier**

Soit  $f(x,y)$  définie par une matrice d'échantillons régulièrement espacés:

N: le nombre de lignes, M: nombre de colonnes

- La T.F est définie par:

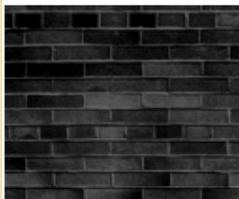
$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp(-j2\pi(ux/M + vy/N))$$

- La T.F inverse est définie par:

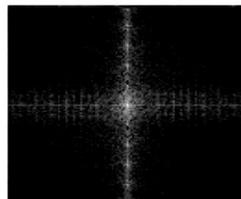
$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp(j2\pi(ux/M + vy/N))$$

Sami Gazzah 2009

80



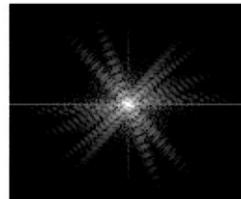
a. Image 1



b. TF de l'image 1



c. Image 2



d. TF de l'image 2

### Compression RLE (Run Length Encoding)

2	2	2	2	128	128	128	120	120	80	96	96	94	94	94	95	97	97	97
---	---	---	---	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----



4	2	3	128	2	120	1	80	2	96	3	94	1	95	4	97
---	---	---	-----	---	-----	---	----	---	----	---	----	---	----	---	----

Utile quand il y a des répétitions dans la chaîne à encoder

Exemple : Formats BMP, TIFF, PCX (Paintbrush)

Sami Gazzah 2009

82

## Chapitre 3:

## RESTAURATION D'IMAGES

Sami Gazzah 2009

83

L'acquisition s'accompagne toujours d'une distorsion/dégradation.

Il existe différentes sources de dégradation (bruit) d'une image.

- bruit lié au contexte de l'acquisition : Bougé, mauvaises conditions d'éclairage,...
- bruit lié au capteur : Capteur de mauvaises qualités, mauvaise mise au point, etc...
- bruit lié à l'échantillonnage : Une mauvaise fréquence d'échantillonnage peut introduire dans l'image des points blancs ou noirs connus souvent sous l'appellation « sel et poivre ».
- bruit lié à la nature de la scène : Présence de fumée, de nuage, etc...

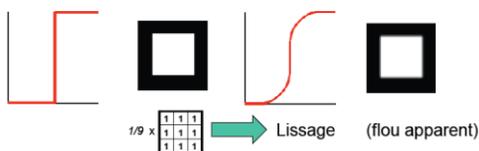
Il faut corriger l'image par un procédé algorithmique.

Sami Gazzah 2009

84



### Filtre moyenneur



Lissage spatial tout en remplaçant la valeur du pixel courant par la moyenne de tous les pixels inclus dans un voisinage centré en ce point  
 Plus le filtre grossit, plus le lissage devient important et plus le flou s'accroît !

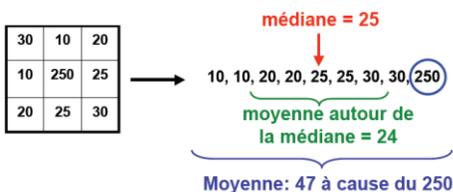
### Autres filtres passe-bas

**Pyramidal:**  $\frac{1}{81} \times \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$       **Conique:**  $\frac{1}{25} \times \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 2 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

**Gaussien:**  $\frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x^2+y^2)}{2\sigma^2}}$        $\frac{1}{98} \times \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 6 & 8 & 6 & 2 \\ 3 & 8 & 10 & 8 & 3 \\ 2 & 6 & 8 & 6 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$

Coef. entiers  
 → calculs plus rapides

### Filtre médian



→ Particulièrement utile pour un bruit de type Poivre & Sel (0 et 255)

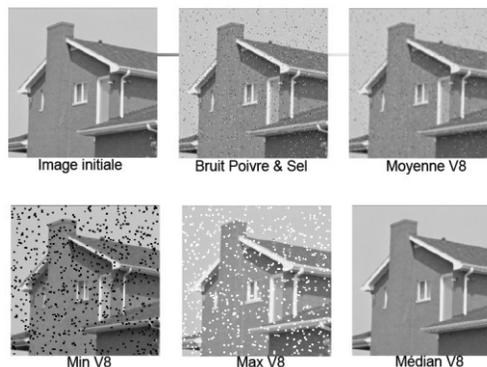
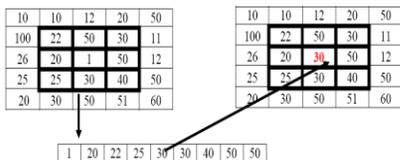
➤ En traitement d'image, les tailles des fenêtres utilisées pour le filtrage médian sont généralement impaires : 3 x 3 ; 5 x 5 ; 7 x 7

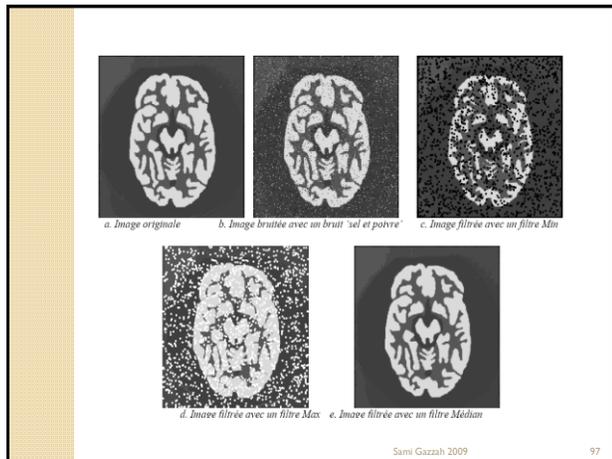
▪ exemple :



➤ Pour les fenêtres de taille paire ( 2 K valeurs ) : après ordonnancement croissant des valeurs, prendre la moyenne des 2 valeurs centrales :

valeur de sortie =  $\frac{(K^{\text{ième}} \text{ valeur ordonnée} + (K+1)^{\text{ième}} \text{ valeur ordonnée})}{2}$





### Comparaison : filtre médian et filtre moyenneur

■ Image « Pièces » de référence, bruitée ( bruit de type impulsionnel ) puis filtrée avec un filtre moyenneur (3×3) et un filtre médian (3×3).

> **Observation :**  
Le filtrage médian est plus adapté que le filtrage linéaire pour réduire le bruit impulsionnel

### Filtre passe-haut

- But :
  - Détecter des contours
  - Réhausser une image
  - Détecter des coins
- Principe: DÉRIVATION: Différence
  - À partir d'un passe-bas
  - Gradient et variantes
  - Laplacien
  - Filtre plus complexe
    - Canny (détecteur de contour)
    - Harris (détecteur de coin)

Sami Gazzah 2009    99

### Filtre passe-haut

- Détection de contours à partir d'un filtre passe-bas

$$I_{PH} = I - I_{PB}$$

Image			Passe-bas			Passe-haut		
0	0	0	$\frac{1}{9}$	1	1	$\frac{1}{9}$	-1	-1
0	1	0		1	1		-1	8
0	0	0		1	1		-1	-1

Sami Gazzah 2009    100

### Filtres différentiels

$f(x, y) = e^{-x^2/2}$

$\frac{\partial f}{\partial x}$     Dérivée première

$\frac{\partial^2 f}{\partial x^2}$     Dérivée seconde

Sami Gazzah 2009    101

## Chapitre 4:

# REHAUSSEMENT D'IMAGES

Sami Gazzah 2009    102

## Introduction

- Le prétraitement d'image est l'ensemble des opérations effectuées sur une image, soit pour l'améliorer, soit pour la restaurer.
- Le prétraitement d'une image vise soit à **renforcer la ressemblance** des pixels, soit au contraire à **accentuer les différences** de pixels d'une même région.
- Accroître la séparabilité des régions composant une scène revient à **augmenter les contrastes** vu.
- On applique le rehaussement des images afin de **faciliter l'interprétation** visuelle et la compréhension des images.
- Pour comprendre ce type de rehaussement, il faut comprendre le concept d'histogramme

Sami Gazzah 2009

103

## Définition de l'histogramme d'une image

- L'histogramme est un outil qui permet de comptabiliser le nombre d'occurrences de chacune des valeurs dans une image en niveaux de gris.
- Dans une image couleur, il est nécessaire d'utiliser plusieurs histogramme sur chacune des composantes du système de représentation de la couleur.

□ Soit  $h$  l'histogramme de l'image  $I$  de dimension  $(w \times h)$ .  $h[i]$  représente le nombre d'occurrence de chaque niveau de quantification  $i$ . La probabilité estimée du niveau  $i$  dans une image  $I$  est :

$$p(i) = \frac{h[i]}{w \cdot h}$$

Sami Gazzah 2009

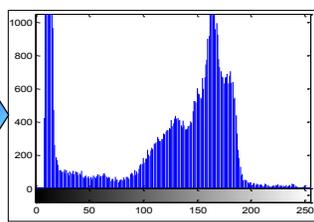
104

## Exemple d'histogramme pour une image en niveaux de gris

- Pour chaque niveau de gris, on compte le nombre de pixels s'y référant
- Pour chaque niveau, on trace le graphe en bâton du nombre de pixels (possibilité de regrouper les niveaux proches en une seule classe)



Image de  $256 \times 256$  pixels, codés chacun sur 8 bits



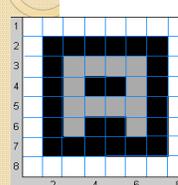
Population de pixels pour chaque niveau de gris [0 ; 255]

## Exemple simple de calcul d'histogramme pour une image

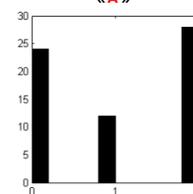
Image « A » en niveaux de gris

Matrice des valeurs de luminance des pixels de l'image « A »

Histogramme de l'image « A »



2	2	2	2	2	2	2	2
2	0	0	0	0	0	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	0	0	0	0	0	2
2	2	2	2	2	2	2	2



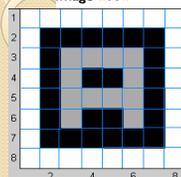
- L'image « A » comporte 3 niveaux de gris différents : 0, 1 et 2.
- Compter le nombre de pixels pour chaque niveau de gris, à l'aide de la matrice des valeurs de luminance.
- Les niveaux 0, 1 et 2 sont respectivement représentés par 24, 12 et 28 pixels  $\Rightarrow$  représentation de cette population de pixels sur l'histogramme.

## Histogramme cumulé d'une image

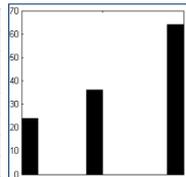
Image « A »

Valeurs de luminance de « A »

Histogramme cumulé de « A »

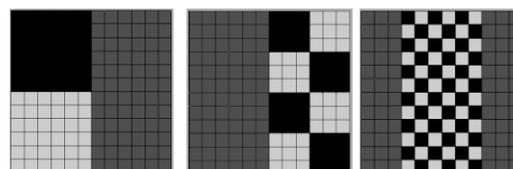


2	2	2	2	2	2	2	2
2	0	0	0	0	0	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	0	0	0	0	0	2
2	2	2	2	2	2	2	2



- Calcul d'un histogramme particulier faisant appel aux cumuls des niveaux de gris  $\Rightarrow$  **Histogramme cumulé**.
- Chaque bâton cumule le nombre de pixels du niveau de gris concerné et des niveaux de gris inférieurs : les niveaux 0, 1, 2 sont donc représentés respectivement par 24, 36 et 64 pixels.
- Utile pour certains traitements d'image tels que **l'égalisation d'histogramme** ( $\Rightarrow$  amélioration de contraste).

## L'histogramme ne donne pas d'informations sur la répartition



Trois images différentes ayant un même histogramme

## Modification de l'histogramme

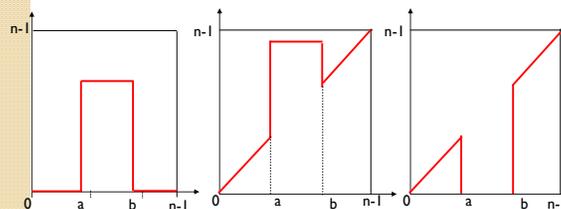
- On appelle dynamique des niveaux de gris l'intervalle  $[n_{min}, n_{max}]$  de niveaux de gris, dans lequel sont compris tous les pixels de l'image.
- Le but de la modification d'histogramme : modifier chaque niveau de gris pour accroître le contraste. On modélise cette transformation par l'équation  $I'(i,j) = g(I(i,j))$ , où  $g$  est la fonction de contraste.
- Chaque transformation sera décrite par une **courbe représentant la relation** entre un niveau de gris de l'image de départ (abscisse) et le niveau qui lui correspond dans l'image améliorée (ordonnée).

Sami Gazzah 2009

109

## Masquage de zones : Extraction d'une fenêtre d'intensité (slicing intensité)

- Cette transformation consiste à remplacer un intervalle de niveaux de gris par du blanc ( $n_{max}$ ), par du noir ( $n_{min}$ ), ou par une valeur intermédiaire du niveau de gris



Courbes représentant, de gauche à droite, l'extraction d'une fenêtre d'intensité et le masquage blanc et noir de l'intervalle  $[a,b]$ .

Sami Gazzah 2009

110

## Exemples de Masquage de zones



a. Image originale sans dégradation    b. Image avec des grilles noires et blanches    c. Masquage gris (60) des intervalles  $[0,10]$  et  $[240,255]$



c. Image avec des grilles noires et blanches    d. Masquage gris des intervalles  $[0,15]$  et  $[240,255]$

Sami Gazzah 2007

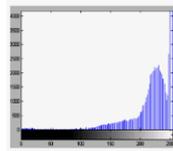
111

## Inversion de la dynamique

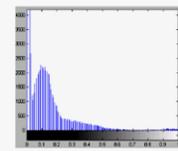


a. Image originale

b. Image inversée



c. Histogramme de l'image originale

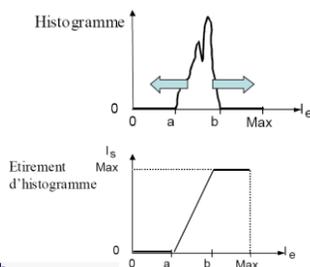


d. Histogramme de l'image inversée

Sami Gazzah 2007

112

## Étirement d'histogramme



- $y = a * x + b$
- $x = \text{min} \rightarrow y = 0$
- $x = \text{max} \rightarrow y = 255$
- $y = 255 * (x - \text{min}) / (\text{max} - \text{min})$

Sami Gazzah 2009

113

## Étirement d'histogramme



Augmentation du contraste par étirement d'histogramme

Sami Gazzah 2009

114

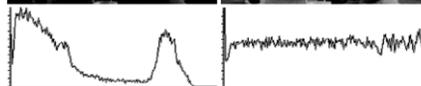
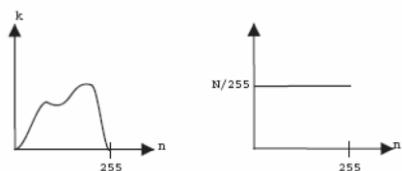
## Égalisation d'histogramme

- **But** : amélioration du contraste
- Distribution **uniforme** des niveaux de gris
- Elle se définit à l'aide de l'histogramme cumulé de l'image  $h_c(n)$  qui représente le nombre de pixels de l'image dont le niveau de gris est inférieur ou égal à  $n$  :  $h_c(n) = \sum_{j=0}^n h(j)$
- Définition de la transformation d'égalisation d'histogramme : soit  $I$  le niveau de gris d'un pixel de départ, le niveau de gris de l'image d'arrivée  $I'$  sera :

$$I' = \frac{255}{\text{Nb de pixels}} h_c(I)$$

Sami Gazzah 2009

115



Sami Gazzah 2009

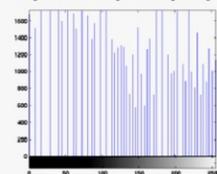
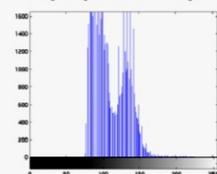
116

## Effet de l'égalisation d'histogramme



Image originale et son histogramme

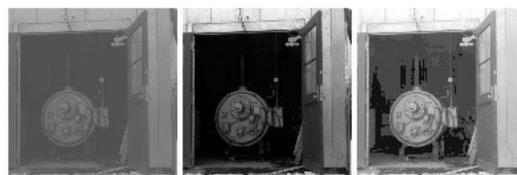
Image et son histogramme après égalisation



Sami Gazzah 2009

117

## Augmentation du contraste : exemples et comparaison



Original

Étirement d'histogramme

Égalisation

Le rehaussement du contraste est plus marqué avec l'égalisation d'histogramme autorisant la détection de structures situées dans l'ombre.

Sami Gazzah 2009

118

## Chapitre 5:

# SEGMENTATION D'IMAGES

Sami Gazzah 2009

119

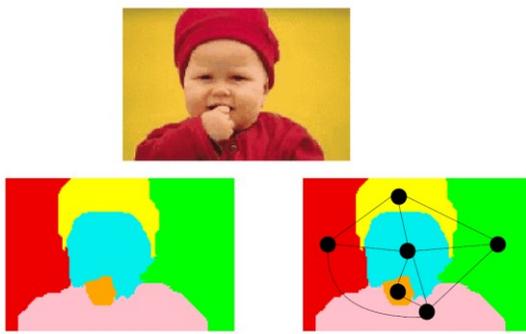
## Introduction

Une image est généralement composée d'objets. L'analyse de cette image nécessite l'**identification** de ces objets.

La segmentation d'images est l'un des problèmes phares du traitement d'images. Elle consiste à **partitionner** l'images en un ensemble de régions connexes. L'intérêt de ces régions est de pouvoir être manipulées ensuite via des traitements de haut niveau pour extraire des caractéristiques de forme, de position, de taille, etc.

Sami Gazzah 2009

120



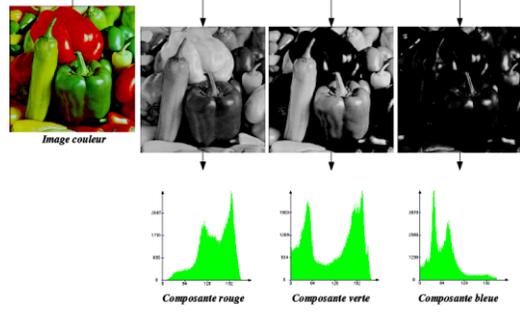
**Segmentation**

**Graphe d'adjacence (Region Adjacency Graph)**

Sami Gazzah 2009 121

### Méthodes statistiques

#### Segmentation par Histogrammes



**Image couleur**

**Composante rouge**

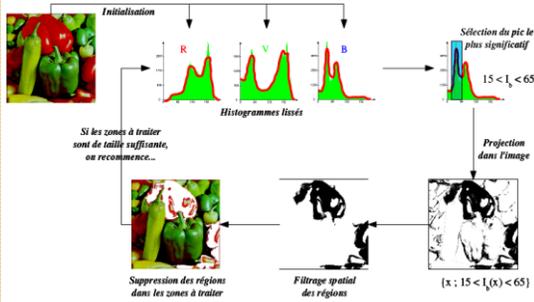
**Composante verte**

**Composante bleue**

Sami Gazzah 2009 122

### Méthodes statistiques

#### Sélection Réursive sur Histogrammes



**Initialisation**

**Histogrammes liés**

**Sélection du pic le plus significatif**

$15 < I_x < 65$

**Projection dans l'image**

**Suppression des régions dans les zones à traiter**

**Filtrage spatial des régions**

$[x; 15 < I_x < 65]$

Olander, Price et Reddy (1978)

Sami Gazzah 2009 123

### Segmentation par approche contour

#### Introduction

Le contour est défini comme le lieu des points connexes qui possèdent une forte **transition d'intensité lumineuse** ou de **texture**. Le traitement qui vise à l'obtenir est appelé extraction de contour; on parle aussi de détection de contour.

Le processus d'extraction de contour peut être décomposé en plusieurs étapes distinctes que nous allons les détailler comme suit :

- La mise en évidence des contours.
- L'extraction des contours.
- La fermeture des contours.

Sami Gazzah 2009 124

### Segmentation par approche contour

#### La mise en évidence des contours

Elle s'obtient par une différentiation de l'image et elle peut s'appliquer à des images binaires ou non, le but de la **détection de contours** est d'obtenir dans un premier temps, une image binaire où chaque contour est d'épaisseur 1 et sans discontinuité:

- rehausser le contour
- suivre le contour

les points du contour correspondent au **maximum en valeur absolue** de la dérivée primaire de l'image, on obtient alors un gradient ou au passage par zéro de la dérivée seconde de l'image, on obtient alors un laplacien

Sami Gazzah 2009 125

### Segmentation par approche contour

#### La mise en évidence des contours

La détection de contours peut être réalisée grâce à des filtres dont les coefficients ont été soigneusement choisis. Nous aborderons dans cette présentation 3 filtres : les filtres de Prewitt, Roberts et Sobel.

Ce jeu de filtres (Prewitt, Roberts et Sobel) est constitué d'une paire de filtres linéaires, qui détectent les contours dans deux directions orthogonales (vertical et horizontal)

Sami Gazzah 2009 126

### Segmentation par approche contour

#### Filtre de Perwitt

Les filtres de Perwitt sont les suivants:

Filtre horizontal  $h = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$

Filtre vertical  $v = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$

Sami Gazzah 2009

127

### Segmentation par approche contour

#### Filtre de Perwitt

A partir d'une image  $I$ , on calcule  $G_h$  et  $G_v$  images correspondant au filtrage de  $I$  par  $h$  et  $v$ . Ces images contiennent les contours horizontaux et verticaux.

Si l'on souhaite obtenir une image de contour unique ne tenant pas compte de l'orientation on calcule :  $G = \sqrt{G_h^2 + G_v^2}$

Orientation du contour :  $\theta = \text{arctg}(G_h + G_v)$

Image binaire...choisir un seuil de détection.

Sami Gazzah 2009

128

### Segmentation par approche contour

#### Filtre de Sobel

Les filtres de Sobel sont les suivants:

Filtre horizontal  $h = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$

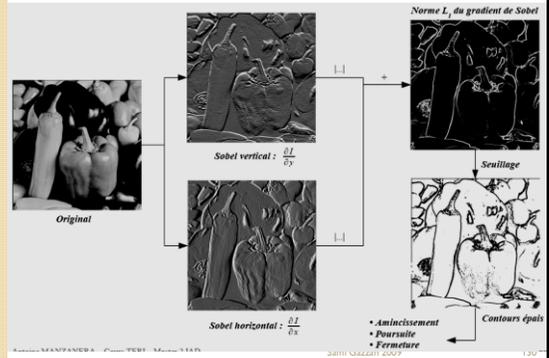
Filtre vertical  $v = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$

Sami Gazzah 2009

129

### Segmentation par approche contour

#### Filtre de Sobel

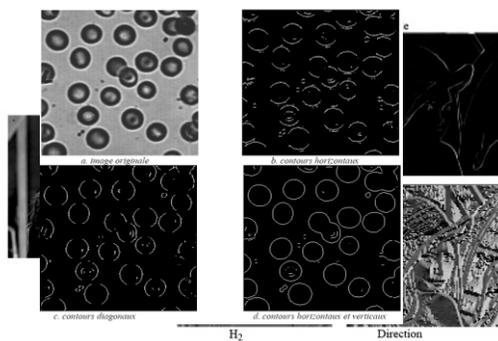


Sami Gazzah 2009

130

### Segmentation par approche contour

#### Filtre de Sobel



Sami Gazzah 2009

131

### Segmentation par approche contour

#### Filtre de Roberts

Les filtres de Roberts sont les suivants:

Filtre diagonal  $h = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

Filtre anti-diagonal  $v = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$

Sami Gazzah 2009

132

## Segmentation par approche contour

### Filtre de Kirsch

Ce détecteur utilise huit noyaux de convolution lui permettant de détecter

des contours en fonction de huit directions: à

$$d1 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} / 15 \quad d2 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} / 15 \quad d3 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} / 15$$

$$d4 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} / 15 \quad d5 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} / 15 \quad d6 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ 5 & 5 & 5 \end{pmatrix} / 15$$

$$d7 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} / 15 \quad d8 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} / 15$$

133

## Segmentation par approche contour

### Contours et Laplacien

Il s'agit de calculer la **dérivée seconde** de l'image.

Le Laplacien est très **sensible au bruit**, pour limiter l'effet du bruit, le

Laplacien est approximé par un **masque** qui effectue en même temps

une opération de filtrage par une Gaussienne qui représente l'étape

de lissage de l'image, puis vient l'application du laplacien. Les masques

du Laplacien Gaussien sont présentés ci-après :

Sami Gazzah 2009

134

## Contours et Laplacien

$$h1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad h2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad h3 = \begin{pmatrix} -1 & -2 & -1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Le contour est ensuite déterminé par la détection des

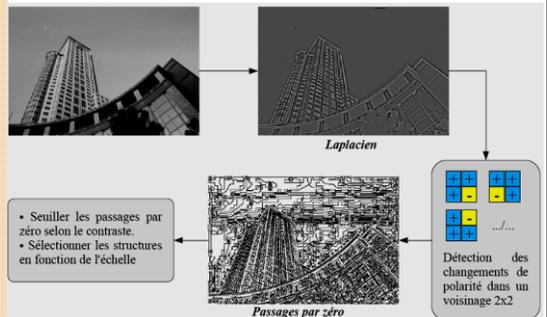
**passages par zéro** du laplacien de l'image

8888888888888888

Sami Gazzah 2009

135

## Contours et Laplacien



Sami Gazzah 2009

136

## Seuillage

Les filtres précédents nous fournissent des images des **sauts** ou **différences d'intensité**.

- Le seuillage permet d'éliminer des points en lesquels la différence d'intensité est très faible.

- Le seuillage peut être poussé afin d'obtenir une image binaire (TOR).

- Le seuillage, dans sa forme la plus simple, ne tient pas compte de la topologie et peut conserver un point bruité isolé ou éliminer une portion de contour d'intensité faible.

Sami Gazzah 2009

137

## Seuillage par hystérésis

c'est une fonction à **deux seuils** : un seuil bas et un seuil haut.

- Tout point dont le module de contour est au-dessus du seuil-haut est un point de contour.

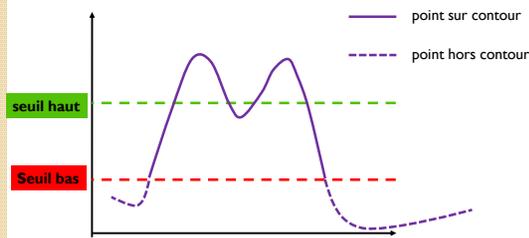
- Tout point dont le module de contour est au-dessous du seuil-bas n'est pas un point de contour.

- Les points de module entre le seuil-bas et le seuil-haut sont des points de contour s'il possède un point de contour dans leur voisinage.

Sami Gazzah 2009

138

## Seuillage par hystérésis



Seuillage par hystérésis : les points de l'image de contour de valeur supérieur au seuil haut sont des points de contours. A partir de ces points de contours, le voisinage de la direction du contour sont examinés : s'ils sont de valeur supérieure au seuil bas, ils sont ajoutés au points de contours

Sami Gazzah 2009

139

## Chapitre 6:

# Introduction aux opérateurs morphomathématique

Sami Gazzah 2009

140

### I. Introduction

- Les opérateurs morphomathématique (=Morphologie Mathématique) permettent d'améliorer les images de contours et apporter d'autres fonctionnalités.
- Ils ont été initialement appliqué sur des images N&B (Matheron et Sarra, 1965), ensuite étendus à des images en niveaux de gris par Dougherty (1978).
- Pour les appliquer sur des images couleurs, on peut les appliquer séparément à chaque composante couleur et combiner les résultats.

Sami Gazzah 2009

141

### I. Introduction

Différentes opérations :

- érosion, dilatation
- ouverture, fermeture
- Squelettisation
- ...

Sami Gazzah 2009

142

### I. Introduction

#### ■ Dilatation

- Ajouter un pixel sur les pourtours de la forme
- boucher les trous

#### ■ Erosion

- Enlever un pixel sur les pourtours de la forme.
- les objets de taille inférieure à celle de l'élément structurant vont disparaître
- agrandit les trous

Sami Gazzah 2009

143

### II. Rappels sur les ensembles

Dans ce qui suit les lettres majuscules désignent des ensembles et les lettres minuscules désignent des éléments appartenant aux ensembles.

Sur les ensembles on définit les opérations suivantes :

- L'égalité:  $X=Y \Leftrightarrow \{x/x \in X \Rightarrow x \in Y \text{ et } x \in Y \Rightarrow x \in X\}$
- L'inclusion:  $X \subseteq Y \Leftrightarrow \{x/x \in X \Rightarrow x \in Y\}$
- L'intersection:  $X \cap Y \Leftrightarrow \{x/x \in X \text{ et } x \in Y\}$
- L'union:  $X \cup Y \Leftrightarrow \{x/x \in X \text{ ou } x \in Y\}$
- La différence:  $X - Y \Leftrightarrow \{x/x \in X \text{ et } x \notin Y\}$
- La différence symétrique:  $X \Delta Y \Leftrightarrow X \cup Y - X \cap Y$
- La complémentarité:  $X^c \Leftrightarrow \{x/x \in E \text{ et } x \notin X\}$  E étant un ensemble servant de référentielle

Sami Gazzah 2009

144

### II. Rappels sur les ensembles

(a) (b)  $A \cup B$  (c)  $A \cap B$   
 (d)  $A'$  (e)  $A - B$

Sami Gazzah 2009 145

### II. Rappels sur les ensembles

- Lois de De Morgan :  $(X \cap Y)^c = X^c \cup Y^c$   
 $(X \cup Y)^c = X^c \cap Y^c$
- Réflexion : La réflexion d'un ensemble A est définie par  $-A = \{-a : a \in A\}$

A -A B -B

Un ensemble symétrique vérifiant  $-A=A$ . C'est le cas des cercles et des carré par exemple

Sami Gazzah 2009 146

### II. Rappels sur les ensembles

- Addition de Minkowski**: l'addition d'un élément x à un ensemble A par x, est définie par:  
 $B = A + x = \{a + x : a \in A\}$  le *translaté* de X par b.

Cette notion n'est pas uniquement ensembliste: elle nous oblige à définir un point de référence

On en déduit l'addition d'un ensemble B à un ensemble A: c'est le résultat de l'addition de chaque élément de B à A.

$$C = A \oplus B = \bigcup (A + b : b \in B) = \bigcup (a + B : a \in A) = \{x : (-B + x) \cap A \neq \emptyset\}$$

L'addition de Minkowski est commutative. Elle est également appelé **dilatation**

Sami Gazzah 2009 147

### II. Rappels sur les ensembles

- Élément structurant**: les opérations morphomathématiques sont définis à partir des ensemble B discrets appelé **élément structurant**. Ces élément permettent de définir le type de voisinage souhaité

A B  $A \oplus B$

Addition de Minkowski d'un ensemble A et d'un ensemble B

Sami Gazzah 2009 148

### III. Dilatation

- Le résultat de **dilatation** de l'ensemble A par l'ensemble B est l'**addition de Minkowski**: c'est l'ensemble des points tel que lorsque B est centré sur un de ces points il y a une intersection non vide entre A et B.
- Pour des images en **niveaux de gris**, on obtient l'image dilatée de la forme A en remplaçant tout pixel par le **maximum** des intensités des **voisins** selon la forme B.

Sami Gazzah 2009 149

### III. Dilatation

#### I. Image N&B

Soit k l'élément structurant de taille  $(2k+1) \times (2k+1)$ . L'algorithme de dilatation est le suivant: on parcourt tous les pixels de l'image sauf les bord de l'image d'épaisseur k. pour chaque pixel du fond rencontré, s'il possède un voisin, au sens de l'élément structurant, qui appartient à un objet, il prend le couleur de l'objet; si non il n'est pas modifié.

Sami Gazzah 2009 150

### III. Dilatation

#### I. Image niveaux de gris

➤ la dilatation (notée  $\oplus$ ) d'une image monochrome «  $f$  » par un élément structurant «  $B$  » est définie en terme de sous-graphe :

- $Y = U(f) \oplus U(B)$
- $f \oplus B = T[Y] = \text{Sup}_l \{ l \text{ tel que } (m, n, l) \in Y \}$

➤ En considérant un élément structurant plan de valeur nulle sur son support :

↳  $(f \oplus B)(P) = \text{Max} \{ \text{valeurs des pixels du voisinage du pixel P} \}$

### III. Dilatation

#### I. Image niveaux de gris

▪ exemple :

n		
115	91	77
95	68	90
55	151	210

Image de référence

$\oplus$

-	-	-
-	*	-
-	-	-

Élément structurant (8-connectivité)

### Exemples de filtrages morphologiques

(élément structurant 3x3 complet)

➔

Image de référence
Image dilatée morphologiquement

➤ **Observation :**  
La **dilatation** en niveaux de gris accroît la luminance des pixels entourés de voisins plus lumineux

### VI. Erosion

- L'érosion est définie par:

$$C = A \ominus B = \bigcap (A - b : b \in B) = \{x : (B + x) \subset A\}$$

- On définit la soustraction de Minkowski par :

$$C = A - (-B) = \bigcap (A + b : b \in B)$$

- On a équivalence entre érosion et soustraction de Minkowski si B est symétrique (si  $B = -B$ ).
- D'un point de vue binaire, l'érode de A par B correspond à l'ensemble des points tels que si B est centre sur ce point, B est entièrement inclus dans A.

Sami Gazzah 2009 154

### VI. Erosion

#### I. Image N&B

on parcourt tous les pixels de l'image sauf les bord de l'image d'épaisseur  $k$ . pour chaque pixel du fond rencontré, s'il possède un voisin, au sens de l'élément structurant, qui appartient à un objet, il prend le **couleur du fond**; si non il n'est pas modifié.

Sami Gazzah 2009 155

### VI. Érosion d'une image en niveaux de gris

➤ L'érosion (notée  $\ominus$ ) d'une image monochrome «  $f$  » par un élément structurant «  $B$  » est définie en terme de sous-graphe :

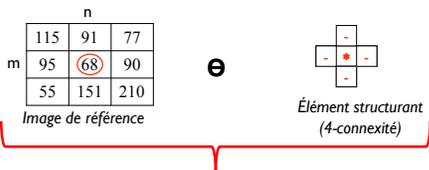
- $Y = U(f) \ominus U(B)$
- $f \ominus B = T[Y] = \text{Sup}_l \{ l \text{ tel que } (m, n, l) \in Y \}$

➤ Classiquement, on considère un élément structurant plan de valeur nulle sur son support

↳  $(f \ominus B)(P) = \text{Min} \{ \text{valeurs des pixels du voisinage du pixel P} \}$

## VI. Érosion d'une image en niveaux de gris

▪ exemple :



liste ordonnée des valeurs du voisinage :

68, 90, 91, 95, 151



$(f \ominus B)(P) = 68$

## Exemples de filtrages morphologiques (élément structurant 3x3 complet)

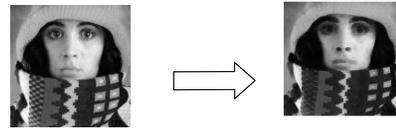


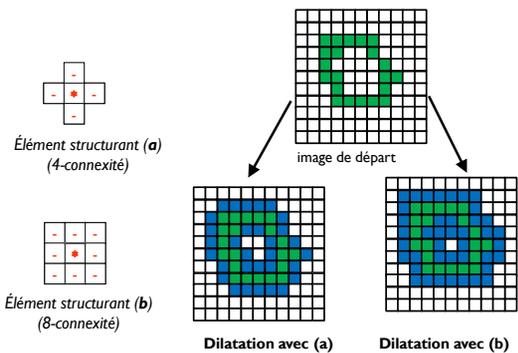
Image de référence

Image érodée

➤ **Observation :**

L'**érosion** en niveaux de gris réduit la luminosité des pixels qui sont entourés de voisins de moindre intensité

**Exercice 1 :** Sur la même image de départ, on réalise deux opérations de dilations avec deux éléments structurant différents



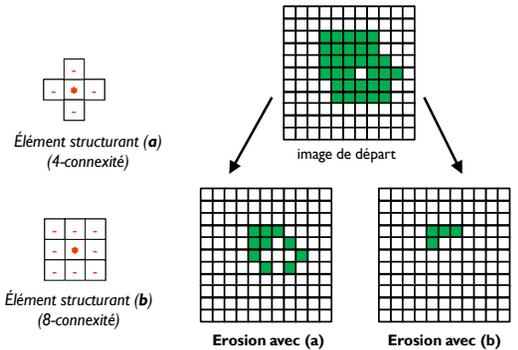
Sami Gazzah 2009 159

**Exercice 1 :**

- cas a : L'élément structurant est à 4-connexité. Chaque pixel du support qui est égal à la valeur 1, ou qui a l'un de ses 4 voisins à la valeur 1 est mis à la valeur 1 après filtrage.
- cas b : L'élément structurant est à 8-connexité. Chaque pixel du support qui est égal à la valeur 1, ou qui a l'un de ses 8 voisins à la valeur 1 est mis à la valeur 1 après filtrage.
- Dans les deux cas, on observe qu'une dilatation élimine les trous isolés dans les objets et dilate le contour des objets en tenant compte de l'élément structurant.

Sami Gazzah 2009 160

**Exercice 2 :** Sur la même image de départ, on réalise deux opérations d'érosions avec deux éléments structurant différents



Sami Gazzah 2009 161

**Exercice 2**

- cas a : L'élément structurant est à 4-connexité (origine à 4 voisins). Chaque pixel du support qui a la valeur 0, ou qui a l'un de ses 4 voisins à la valeur 0 est mis à la valeur 0 après filtrage.
- cas b : L'élément structurant est à 8-connexité. Chaque pixel du support qui a la valeur 0, ou qui a l'un de ses 8 voisins à la valeur 0 est mis à la valeur 0 après filtrage.
- Dans les deux cas, on observe qu'une érosion élimine les pixels isolés sur le fond et érode le contour des objets.

Sami Gazzah 2009 162

## Conclusions

### • Erosion

- les objets reliés entre eux vont être séparés.
- Une érosion de taille  $n$  peut se réaliser en répétant une érosion  $n$  fois avec un élément structurant de taille 1 ou en appliquant une seule érosion avec un élément structurant de taille  $n$ .

### • Dilatation

- des objets sont situés à une distance moins grande que la taille de l'élément structurant, il vont fusionner.

Sami Gazzah 2009

163

## Erosion et dilatation

### Avantages/Utilisation

- Méthode rapide
- Erosion utile pour séparer des structures accolées
- Erosion utile pour « séparer » le fond des structures
- Dilatation utile pour combler des pixels « manquants »
- Souvent utilisé par cycle(s) alternant érosion/dilatation

### Inconvénients

- Très sensible à la forme du kernel : Les structures composant l'image sont déformées et prennent la forme du kernel (carré, octogonal, etc.)

Sami Gazzah 2009

164

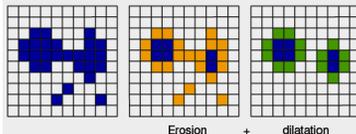
## V. Ouverture et fermeture

### I. Ouverture

L'ouverture: (*opening*) consiste à une érosion suivi d'une dilatation

$$C = A \circ B = (A \ominus B) \oplus B$$

Les petites structures et détail fins disparaissent et ne sont pas recrées lors de la dilatation. Une opération d'ouverture est ainsi intéressante pour éliminer les petits morceaux de contours bruités dans une image de contours.



Elimine pixels isolés sans intérêt

Sami Gazzah 2009

165

## V. Ouverture et fermeture

### II. Fermeture

La fermeture: (*closing*) consiste à une dilatation suivi d'une érosion

Les structures proches fusionnent lors de la dilatation mais seules les fusions ponctuelles disparaissent à nouveau lors de l'érosion. Une opération de fermeture est ainsi intéressante pour boucher les petits trous.



Sami Gazzah 2009

166

## Ouverture et Fermeture

- ★ Lors d'une érosion, on réduit le contour de la structure d'un pixel
- ★ Lors d'une dilatation, on augmente le contour de la structure d'un pixel

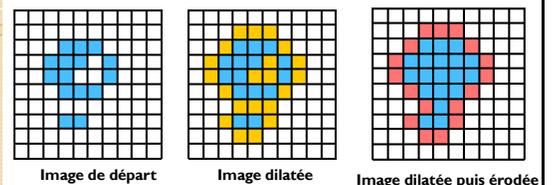
### Utilisation

- ★ Cycles  
nombre de cycles = diamètre maximum de la structure à faire disparaître (ou apparaître)

Sami Gazzah 2009

167

**Exercice 3 :** Sur l'image de départ, on réalise une opération de fermeture binaire par l'élément structurant 4-connexité. Conclusions...



Élément structurant  
(4-connexité)

**Fermeture = dilatation puis érosion**

- ☐ Comblé les trous
- ☐ Rétablir les jonctions
- ☐ ... tout en préservant la taille

Sami Gazzah 2009

168

### Exercice 4

Sur l'image de départ, on réalise une opération d'ouverture binaire par l'élément structurant 4-connectivité. Conclusions...

Image de départ      Image érodée      Image érodée puis dilatée

Élément structurant (4-connectivité)

**Ouverture = érosion puis dilatation**

- Supprimer les fausses alarmes
- Séparer les objets accolés
- ... tout en préservant la taille

Sami Gazzah 2009      169

### Exercice 5

Appliquer les filtres suivants sur l'image donnée. Quelles sont les valeurs choisies aux bords de l'image et quelles sont les modifications visuelles ?

- filtre moyenneur  $M3 \times 3(1/9)$
- érosion
- dilatation
- médian
- fermeture
- ouverture

Sami Gazzah 2009      170

## IV. Amincissement et Squelettisation

La squelette (skeleton) d'une forme est l'axe médian. L'intérêt de cette opération est dans :

- La reconnaissance de caractères.
- Détection des réseaux routiers.
- La planification de trajectoire,...

Une méthode couramment employée est une amincissement répété jusqu'à la convergence (n'apporte aucune modification)

Sami Gazzah 2009      171

## IV. Amincissement et Squelettisation

### 1. Opération Tout Ou Rien «Hit or Miss»

Cette transformation est définie par :  $A \otimes B = (A \ominus B) \cap (A^c \ominus B^c)$

l'élément structurant étendu est centré sur chaque pixel de l'image comme pour les opérations ensemblistes.

la question posée est: est ce que le contenu de l'élément structurant?

B<sup>c</sup> est le complémentaire local de B,

si la réponse est **oui**, le pixel résultant vaut **1**, si **non** il vaut **0**.

Une application permet de détecter des configurations locales particulières dans une image.

plusieurs masques peuvent être utilisés indépendamment, les images résultat étant ensuite combinées par une **opération ou**.

Sami Gazzah 2009      172

## IV. Amincissement et Squelettisation

### 1. Opération «Hit or Miss»

0	0
0	1
1	1

haut gauche

0	0
1	1
1	0

haut droit

1	1
0	1
0	0

bas gauche

1	1
1	0
0	0

bas droit

image originale      coins détectés

Exemple d'opération Hit or Miss, détection des coins

Sami Gazzah 2009      173

## IV. Amincissement et Squelettisation

### 1. Amincissement (thinning)

- La transformation TOR nous permet également de définir l'amincissement :  $thin(A) = A / (A \otimes B)$
- L'amincissement consiste à soustraire l'image résultat d'un hit or miss de l'image initiale.
- des amincissements successifs conduisent au squelettisation.

Sami Gazzah 2009      174

### IV. Amincissement et Squelettisation

#### II. Squelettisation

Appliquer des amincissements successifs puis s'arrêter lorsque l'amincissement n'apporte plus de modification. C'est une érosion particulière où les pixels sont supprimés à la condition qu'ils ne provoquent pas la séparation d'une région en deux.

Sami Gazzah 2009 175

### IV. Amincissement et Squelettisation

#### II. Squelettisation

Sami Gazzah 2009 176

## Chapitre 7:

# Transformations géométriques 2D

Sami Gazzah 2009 177

### I. Introduction

On entend par transformation 2D l'ensemble des transformations du plan pouvant être appliquées aux pixels de l'image, sans considération de l'intensité.

Par exemples :

- Réduction, agrandissement
- Déformation d'image
- Recalage d'une photo scannée
- Recaler des images que l'on veut comparer
- Mosaïque d'images

Sami Gazzah 2009 178

### I. Transformations euclidiennes

- Elles préservent les distances et les angles
- Elles sont les compositions de translations et de rotations

#### 1) Translation

$$\begin{cases} x' = x + T_x \\ y' = y + T_y \end{cases}$$

Où  $T_x$  et  $T_y$  sont les déplacements en  $x$  et en  $y$  de la translation

Sami Gazzah 2009 179

#### 2) Rotation

Rotation autour de l'origine :

$$\begin{cases} x' = x \cos(\theta) - y \sin(\theta) \\ y' = y \cos(\theta) + x \sin(\theta) \end{cases}$$

Où  $\theta$  est l'angle de rotation dans le sens anti-horaire.

Cas général autour d'un point fixe :

$$\begin{cases} x' = x_r + (x - x_r) \cos(\theta) - (y - y_r) \sin(\theta) \\ y' = y_r + (y - y_r) \cos(\theta) + (x - x_r) \sin(\theta) \end{cases}$$

Où  $\theta$  est l'angle de rotation dans le sens anti-horaire.

Sami Gazzah 2009 180

### Rotation - exemples

Sami Gazzah 2009 181

### 3) Remarques (1)

- Les transformations ne sont pas commutatives :
  - Rotation x Translation  $\neq$  Translation x Rotation

Sami Gazzah 2009 182

### 3) Remarques (2)

- Par contre, on peut inverser :
  - 2 rotations:
    - Rotation1 x Rotation2 = Rotation2 x Rotation1
  - 2 translations:
    - Translation1 x Translation2 = Translation2 x Translation1

Sami Gazzah 2009 183

### II. Homothéties

Changement d'échelle par rapport à l'origine (scaling, zoom)

$$\begin{cases} x' = S_x \cdot x \\ y' = S_y \cdot y \end{cases}$$

Où  $S_x$  et  $S_y$  sont les facteurs d'agrandissement ( $>1$ ) ou de réduction ( $<1$ ) en  $x$  et en  $y$ .

Sami Gazzah 2009 184

### II. Homothéties

- Changement d'échelle autour d'un point fixe :

Sami Gazzah 2009 185

### II. Homothéties

- Si on veut changer l'échelle autour d'un point fixe on utilise plutôt une translation combinée avec le changement d'échelle :

$$x' = x_f + (x - x_f) S_x \quad y' = y_f + (y - y_f) S_y$$

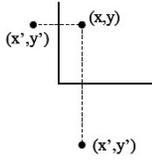
Où  $S_x$  et  $S_y$  sont les facteurs d'agrandissement ( $>1$ ) ou de réduction ( $<1$ ) en  $x$  et en  $y$ .

Sami Gazzah 2009 186

### III. Réflexions

Réflexions par rapport aux axes (flip, miroir)



$$\begin{cases} x' = x \\ y' = -y \end{cases} \text{ et } \begin{cases} x' = -x \\ y' = y \end{cases}$$




Flip      Miroir

Sami Gazzah 2009      187

### IV. Cisaillement

Cisaillement (glissement, "shearing") : on étire suivant un ou plusieurs axes,

Cisaillement selon les abscisses :

$$\begin{cases} x' = x + Sh_x \cdot y \\ y' = y \end{cases}$$

Cisaillement selon les ordonnées :

$$\begin{cases} x' = x \\ y' = y + Sh_y \cdot x \end{cases}$$

Cisaillement général :

$$\begin{cases} x' = x + Sh_x \cdot y \\ y' = y + Sh_y \cdot x \end{cases}$$







Sami Gazzah 2009      188

### Transformations inverses

- $T_x$  et  $T_y$        $\Rightarrow$        $-T_x$  et  $-T_y$
- $\theta$        $\Rightarrow$        $-\theta$
- $S_x$  et  $S_y$        $\Rightarrow$        $1/S_x$  et  $1/S_y$
- $Sh_x$  et  $Sh_y$        $\Rightarrow$        $-Sh_x$  et  $-Sh_y$

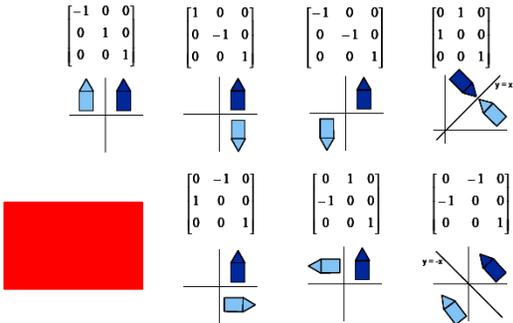
Sami Gazzah 2009      189

### Représentation Matricielle

- Exemples :
  - Rotation       $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$        $\begin{cases} x' = x \cos(\theta) - y \sin(\theta) \\ y' = y \cos(\theta) + x \sin(\theta) \end{cases}$
  - Mise à l'échelle       $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$        $\begin{cases} x' = S_x \cdot x \\ y' = S_y \cdot y \end{cases}$
  - Cisaillement       $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & Sh_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$        $\begin{cases} x' = x + Sh_x \cdot y \\ y' = y \end{cases}$

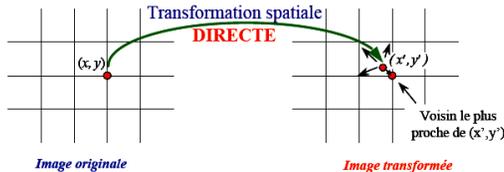
Sami Gazzah 2009      190

### IV. Autres (réflexions, miroirs et flips)




Sami Gazzah 2009      191

### Interpolation Directe



o Difficile de distribuer le niveau de gris sur les pixels voisins

o On n'a pas la certitude que tous les pixels de l'image transformée recevront une valeur => "trous"

Sami Gazzah 2009      192

### Interpolation Inverse

Transformation spatiale  
**INVERSE**

Voisin le plus proche de (x,y)

Image originale Image transformée

- Interpolation de la valeur du pixel à partir des voisins plus facile.
- Aucun "trou" car on parcourt tous les pixels de l'image transformée.

Sami Gazzah 2009 193

### V. Objet vectoriel - Objet bitmap

**Objet vectoriel**  
Essentiellement, on transforme les sommets (ou points de contrôle) et on retrace...

**Objet "bitmap"**  
On doit calculer la nouvelle position de chaque pixel!! De plus on doit généralement interpoler les niveaux de gris (couleurs)...

Sami Gazzah 2009 196

### IV. Assignation des niveaux de gris

Nécessaire car la position d'un pixel transformé n'est pas entière...

Plusieurs méthodes :

- Interpolation au plus proche voisin
- Interpolation bilinéaire
- Interpolation bicubique
- Autres types d'interpolation

Sami Gazzah 2009 195

### IV. Assignation des niveaux de gris

L'interpolation de l'image, c'est ce qui arrive lorsqu'on redimensionne une image numérique, ou qu'on la redimensionne avec une distorsion, qu'on change la perspective, etc...

Exemple 1:

- Image originale (sans interpolation) : INFOGRAPHIE
- Image agrandie, et donc interpolée :

Perte de netteté, due à l'interpolation.

Sami Gazzah 2009 196

### IV. Assignation des niveaux de gris

Exemple 2: recadrage d'image

Nous allons agrandir cette image à 180% :  
**Il va donc nous manquer des informations, là où l'on va créer des pixels pour agrandir l'image**

Résultat **sans interpolation** : le logiciel agrandit simplement les pixels :

Résultat **avec interpolation** : le logiciel calcule une couleur intermédiaire pour chaque nouveau pixel

Sami Gazzah 2009 197

### IV. Assignation des niveaux de gris

Grey level represented by a digital number 88 44

Camera track Physical landscape

44 48 66 84 88 66 44

RECORDED DATA  
Pixel values returned after averaging for the scanned area

Image originale

Interpolation bicubique Interpolation bilinéaire Interpolation au plus proche voisin

Sami Gazzah 2009 198

### IIV. Interpolation au plus proche voisin

- C'est l'algorithme le plus rapide à calculer,
- Prend en compte dans le calcul que le pixel le plus proche du point à interpoler.
- Consiste tout simplement à agrandir chaque pixel.

**Exemple :** Interpolation du plus proche voisin par copie des pixels :  
Copie de chaque colonne et de chaque rang



Sami Gazzah 2009

199

### IIV. Interpolation bilinéaire

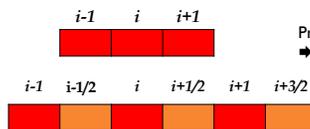
■ Principe 1D

$$y_i = xi$$

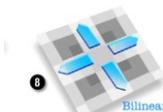
$$Y_{i+1/2} = (x_i + x_{i+1})/2$$

Problème pour le dernier point  
→ Extrapolation linéaire de :

$$2 \cdot x_{i+1} - x_{i-1}$$



■ Fonction bilinéaire de 4 pixels voisins ( en 2 D )



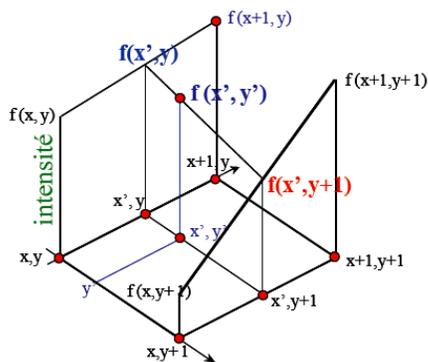
attribuer à chaque point cible une combinaison linéaire des quatre points sources les plus proches par la transformation inverse.

Bilinear interpolation creates new pixels based on color averages from both the horizontal and vertical neighbors of the area to be revised.

Sami Gazzah 2009

200

### IIV. Interpolation bilinéaire



Sami Gazzah 2009

201

### IIV. Interpolation bilinéaire

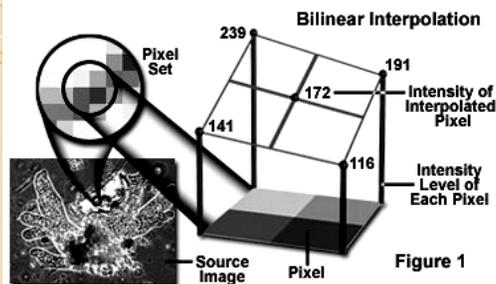
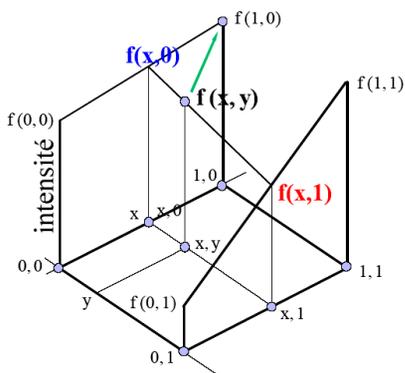


Figure 1

Sami Gazzah 2009

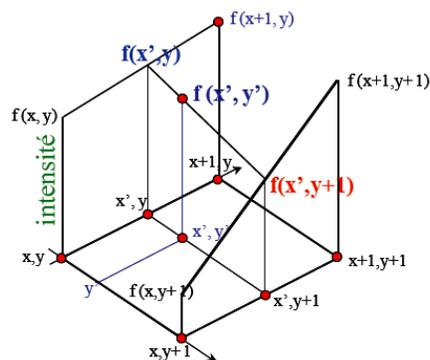
202



Sami Gazzah 2009

203

### IIV. Interpolation bilinéaire



Sami Gazzah 2009

204

### IIV. Interpolation bilinéaire

$f(x,y) = ax + by + cxy + d$

4 inconnues a, b, c, d  $\rightarrow$  4 points

Solution:

$$a = f(x+1,y) - f(x,y)$$

$$b = f(x,y+1) - f(x,y)$$

$$c = f(x+1,y+1) + f(x,y) - f(x,y+1) - f(x+1,y)$$

$$d = f(x,y)$$

Implantation:

$$f(x',y) = f(x,y) + (x'-x)[f(x+1,y) - f(x,y)]$$

$$f(x',y+1) = f(x,y+1) + (x'-x)[f(x+1,y+1) - f(x,y+1)]$$

$$f(x',y') = f(x',y) + (y'-y)[f(x',y+1) - f(x',y)]$$

### IIV. Interpolation bilinéaire

Relations dans les triangles :  $A / B = C / D$

$$(x'-x) / (f(x',y) - f(x,y)) = (x+1-x') / (f(x+1,y) - f(x',y))$$

$$\Rightarrow (x'-x)(f(x+1,y) - f(x,y)) + f(x,y) = f(x',y)$$

etc...

### IIV. Interpolation bilinéaire : exemple

100	150	y+1
175	200	y
x	x+1	

Quel est la valeur interpolée en  $(x+0.7, y+0.2)$ ?

Sami Gazzah 2009 207

### IIV. Interpolation bicubique

**Principe 1D**

- Même raisonnement que précédemment sauf que  $y_{i+1/2} = (-x_{i+1} + 9x_i + 9x_{i-1} - x_{i-2}) / 16$
- Problèmes pour  $y_{i/2}$ , ainsi que pour les 2 derniers points
  - $\Rightarrow$  2 extrapolations :
    - une parabolique à distance 1  $x_N = 3x_{N-1} - 3x_{N-2} + x_{N-3}$
    - une linéaire à distance 2  $x_{N+1} = 2x_N - x_{N-1}$
- Le facteur zoom peut être plus élevé, par exemple 3, ce qui donne pour une interpolation cubique :
 
$$y_{i+1/3} = (-5x_{i-1} + 60x_i + 30x_{i+1} - 4x_{i+2}) / 81$$

$$y_{i+2/3} = (-4x_{i-1} + 30x_i + 60x_{i+1} - 5x_{i+2}) / 81$$

Sami Gazzah 2009 208

### exemple

Changement d'échelle x 2

Plus proche voisin      Bilineaire      Bicubique

Sami Gazzah 2009 209

## Chapitre 8:

# Génération de segments de droite et de cercles en 2D

Sami Gazzah 2009 210

# I. Segments 2D, cercles

## 1. Introduction

- Infographie 2D → besoin de dessiner des primitives de base (segments, cercles, ...)
- Espace discrétisé (pixels) → il faut trouver un moyen pour afficher des segments ou des cercles

■ Comment trouver les pixels à allumer pour approximer au mieux une droite ou un cercle ?

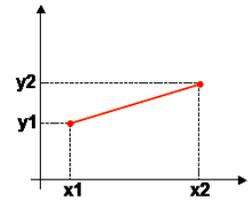
■ Algorithme rapide et précis

# I. Segments 2D, cercles

## 2. Génération de segments de droite 2D

- À partir des coordonnées des 2 extrémités d'un segment...
- Équation d'une droite :  $y=mx+b$
- Calcul de la pente

$$a = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$



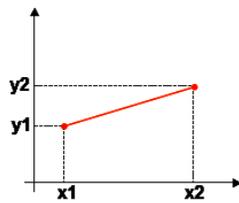
# I. Segments 2D, cercles

## 2. Génération de segments de droite 2D

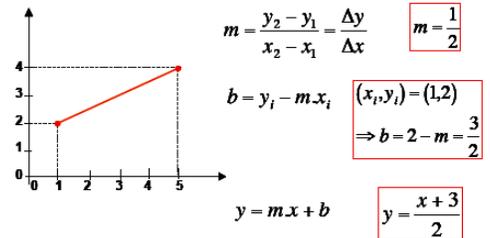
Rappel....

- À partir des coordonnées des 2 extrémités d'un segment...
- Équation d'une droite :  $y=mx+b$
- Calcul de la pente

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$



## Exemple



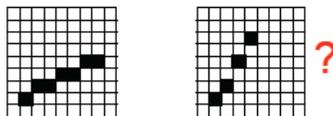
# I. Segments 2D, cercles

## 2. Génération de segments de droite 2D

- À partir des coordonnées des 2 extrémités d'un segment...
- Un algorithme simple :  
 Boucle de  $x_1$  à  $x_2$ ,  $dx=1$ ,  
 ALLUME ( $x$ , Arrondir( $y$ ))

Avec  $x_2 > x_1$

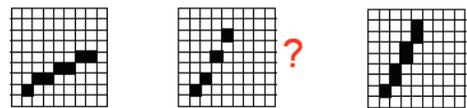
Arrondir( $y$ ) = Arrondir( $ax+b$ )



# I. Segments 2D, cercles

## 2. Génération de segments de droite 2D

Choix de l'axe...



Si  $|m| \leq 1$   
 Boucle de  $x_1$  à  $x_2$ ,  $dx=1$ ,  
 ALLUME ( $x$ , Arrondir( $mx+b$ ))

Si  $|m| > 1$   
 Boucle de  $y_1$  à  $y_2$ ,  $dy=1$ ,  
 ALLUME (Arrondir( $(y-b)/m$ ),  $y$ )

$y=mx+b \Rightarrow x=(y-b)/m$

→ Peu efficace :  
 • à chaque itération : 1 multiplication (ou division) + 1 addition + 1 arrondi.

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

**DDA (Digital Differential Analyzer)**

$|m| < 1 \Rightarrow$  incrémente  $x$   
 $x_{i+1} = x_i + 1$   
 $y_{i+1} = mx_{i+1} + b$   
 $= m(x_i + 1) + b$   
 $= y_i + m$

**Si  $|m| < 1$**   
 Boucle de  $x1$  à  $x2$ ,  $dx=1$   
 $y = y + m$   
 ALLUME  $(x, \text{Arrondir}(y))$

**Si  $|m| > 1$**   
 Boucle de  $y1$  à  $y2$ ,  $dy=1$   
 $x = x + 1/m$   
 ALLUME  $(\text{Arrondir}(x), y)$

*y et m doivent être des réels*

Sami Gazzah 2009 217

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

**Algorithme de Bresenham**

- Calcul en entier ! (donc plus rapide)
- Basé sur le choix entre deux points discrets, dépendant de leur distance respective avec le point réel idéal.
- On simplifie le problème en traitant d'abord le tracé d'un segment dans le 1<sup>er</sup> octant  $\rightarrow m$  inférieur à 1 et positif ( $0 < m < 1$ )

Facilement généralisable aux autres octants

Sami Gazzah 2009 218

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

**Algorithme de Bresenham**

- Tracé d'un segment dans le premier octant  $m < 1$
- On part d'un point  $(x_i, y_i)$  allumé :

$y = m(x_i + 1) + b$   
 $d_1 = y - y_i$   
 $= m(x_i + 1) + b - y_i$   
 $d_2 = (y_i + 1) - y$   
 $= y_i + 1 - m(x_i + 1) - b$

Sami Gazzah 2009 219

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

**Algorithme de Bresenham**

- On ne calcule pas réellement les distances  $\Rightarrow$  on vérifie si la différence entre les deux distances est positive ou négative :

$d_1 = m(x_i + 1) + b - y_i$   
 $d_2 = y_i + 1 - m(x_i + 1) - b$   
 $d_1 - d_2 = 2m(x_i + 1) - 2y_i + 2b - 1$

$d_1 > d_2 \Rightarrow ?$   
 $d_1 < d_2 \Rightarrow ?$

Le signe de la longueur  $(d_1 - d_2)$  détermine s'il faut allumer le pixel L ou H.

Sami Gazzah 2009 220

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

**Algorithme de Bresenham**

- Comme  $m = \Delta y / \Delta x$ , on multiplie cette différence par  $\Delta x$  pour éviter la division:

$d_1 - d_2 = 2m(x_i + 1) - 2y_i + 2b - 1$

$p_i = \Delta x(d_1 - d_2)$   
 $= \Delta x \left( 2 \frac{\Delta y}{\Delta x} x_i + 2 \frac{\Delta y}{\Delta x} - 2y_i + 2b - 1 \right)$   
 $= 2\Delta y x_i - 2\Delta x y_i + c$   
 avec  $c = 2\Delta y + \Delta x(2b - 1)$

Mais que vaut  $p_1$  ?

Sami Gazzah 2009 221

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

**Algorithme de Bresenham**

$p_1 = 2\Delta y x_i - 2\Delta x y_i + 2\Delta y + \Delta x(2b - 1)$

avec  $b = y_i - m x_i$  et  $m = \frac{\Delta y}{\Delta x}$

$p_1 = 2\Delta y x_i - 2\Delta x y_i + 2\Delta y + \Delta x \left( 2y_i - 2 \frac{\Delta y}{\Delta x} x_i - 1 \right)$

$p_1 = 2\Delta y x_i - 2\Delta x y_i + 2\Delta y + 2\Delta x y_i - 2\Delta y x_i - \Delta x$

$p_1 = 2\Delta y - \Delta x \Rightarrow$  ENTIER

Sami Gazzah 2009 222

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

Algorithme de Bresenham

Pour éviter de calculer cette différence pour chaque point, on cherche une façon de calculer les différences subséquentes par incrémentation.

$$p_i = 2\Delta y \cdot x_i - 2\Delta x \cdot y_i + c \quad \text{avec } c = 2\Delta y + \Delta x(2b-1)$$

$$p_{i+1} = 2\Delta y \cdot x_{i+1} - 2\Delta x \cdot y_{i+1} + c$$

$$p_{i+1} - p_i = 2\Delta y(x_{i+1} - x_i) - 2\Delta x(y_{i+1} - y_i)$$

$$p_{i+1} = p_i + 2\Delta y - 2\Delta x(y_{i+1} - y_i)$$



SAMI GAZZAH 2009

223

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

Algorithme de Bresenham

allume( $x_i, y_i$ );

$$p_{i+1} = p_i + 2\Delta y - 2\Delta x(y_{i+1} - y_i)$$

$p_i = 2\Delta y - \Delta x$ ;

boucle de  $x_1$  à  $x_2$ ;

si  $p_i < 0$  alors  $y_{i+1} = y_i$ ;

$p_{i+1} = p_i + 2\Delta y$ ;

sinon  $y_{i+1} = y_i + 1$ ;

$p_{i+1} = p_i + 2(\Delta y - \Delta x)$ ;

fin - si

allume( $x_{i+1}, y_{i+1}$ );

fin - boucle

Note: 1<sup>er</sup> quadrant seulement

$$p_i = \Delta x(d_1 - d_2)$$

SAMI GAZZAH 2009

224

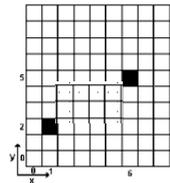
### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

Exemple:

Tracer le segment de (1,2) à (6,5)

```
allume(x1,y1);
p1 = 2Δy - Δx;
boucle de x1 à x2;
si p1 < 0 alors y1+1 = y1;
                    p1+1 = p1 + 2Δy;
sinon y1+1 = y1 + 1;
                    p1+1 = p1 + 2(Δy - Δx);
fin - si
allume(x1+1,y1+1);
fin - boucle
```



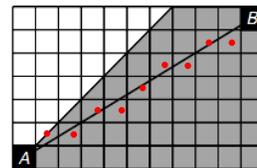
SAMI GAZZAH 2009

225

#### Exercice

Tracer le segment de droite

$p_i$     $x_i$     $y_i$



$$\Delta x = 10$$

$$\Delta y = 6$$

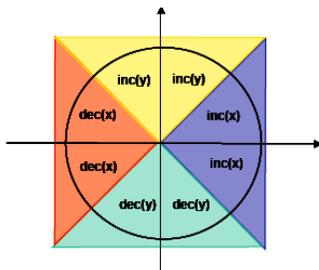
SAMI GAZZAH 2009

### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

Algorithme de Bresenham

Techniques de généralisation aux autres octants:



SAMI GAZZAH 2009

227

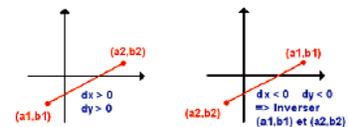
### I. Segments 2D, cercles

#### 2. Génération de segments de droite 2D

Algorithme de Bresenham

Techniques de généralisation aux autres octants:

... interchanger les points ...



Si  $|\Delta x| < |\Delta y|$ , on interchange le rôle de x et y.

SAMI GAZZAH 2009

228

### I. Segments 2D, cercles

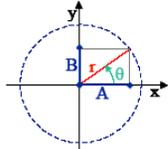
#### 3. Génération de cercles

##### Rappels (1)

- Équation cartésienne d'un cercle :

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

- Coordonnées polaires



$$\cos(\theta) = \frac{A}{r} \Rightarrow A = r \cdot \cos(\theta)$$

$$\sin(\theta) = \frac{B}{r} \Rightarrow B = r \cdot \sin(\theta)$$

- Équation paramétrique (polaire) d'un cercle :

$$x = x_c + r \cdot \cos(\theta) \quad y = y_c + r \cdot \sin(\theta)$$

229

### I. Segments 2D, cercles

#### 3. Génération de cercles

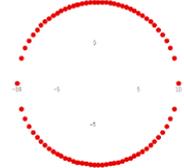
##### Rappels (2)

- Équation cartésienne d'un cercle :

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \Rightarrow y = y_c \pm \sqrt{r^2 - (x - x_c)^2}$$

- Algorithme simple :

Boucle de  $x = -r$  à  $r$ , pas= $\delta x$ ,  
 ALLUME ( $x_c+x$ ,  $y_c+y$ )  
 ALLUME ( $x_c-x$ ,  $y_c-y$ )



- Problèmes:

- L'espace des points n'est pas uniforme (trous possibles)
- Calculs importants (racine carrée)

Sami Gazzah 2009

230

### I. Segments 2D, cercles

#### 3. Génération de cercles

##### Rappels (3)

- Équation paramétrique (polaire) d'un cercle :

$$x = x_c + r \cdot \cos(\theta) \quad y = y_c + r \cdot \sin(\theta)$$

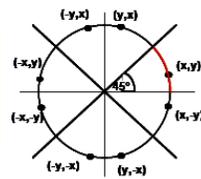
Boucle de  $\theta = 0$  à  $45^\circ$ , pas= $\delta\theta$ ,

ALLUME ( $x+x_c$ ,  $y+y_c$ )

ALLUME ( $x+x_c$ ,  $-y+y_c$ )

ALLUME ( $y+y_c$ ,  $x+y_c$ )

...



- Remarques:

- Calculs toujours importants (fonction trigo.)
- 60 grand  $\rightarrow$  on complète avec segments de droite

Sami Gazzah 2009

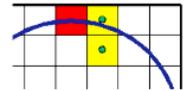
231

### I. Segments 2D, cercles

#### 3. Génération de cercles

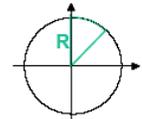
##### Algorithme de Bresenham

- Toujours basé sur le choix entre deux points discrets, dépendant de leur distance respective avec le point réel.



- On travaille dans le 2e octant à partir de (0,r) ainsi pour un point ( $x_i$ ,  $y_i$ ), le prochain point sera:

$$(x_i + 1, y_i) \text{ ou } (x_i + 1, y_i - 1).$$



Sami Gazzah 2009

232

### I. Segments 2D, cercles

#### 3. Génération de cercles

##### Algorithme de Bresenham

- On calcule le prochain y et les différences respectives avec les points discrets de la façon suivante:

$$y^2 = r^2 - (x_i + 1)^2$$

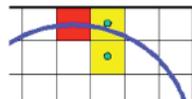
$$d_1 = y_i^2 - y^2 = y_i^2 - r^2 + (x_i + 1)^2$$

$$d_2 = y^2 - (y_i - 1)^2 = r^2 - (x_i + 1)^2 - (y_i - 1)^2$$

$$d_1 - d_2 = 2(x_i + 1)^2 + y_i^2 + (y_i - 1)^2 - 2r^2$$

- Le signe de pi nous dira si nous devons choisir le point latéral ou le diagonal.

$$p_i = (d_1 - d_2) = 2(x_i + 1)^2 + y_i^2 + (y_i - 1)^2 - 2r^2$$



Sami Gazzah 2009

233

### I. Segments 2D, cercles

#### 3. Génération de cercles

##### Algorithme de Bresenham

- Encore une fois, pour éviter de calculer cette différence pour chaque point, on cherche une façon de calculer les différences subséquentes par incrémentation.

$$p_i = (d_1 - d_2) = 2(x_i + 1)^2 + y_i^2 + (y_i - 1)^2 - 2r^2$$

$$p_{i+1} = 2[(x_i + 1) + 1]^2 + y_{i+1}^2 + (y_{i+1} - 1)^2 - 2r^2$$

$$p_{i+1} = 2(x_i + 1)^2 + 2(x_i + 1) + 1 + y_{i+1}^2 + (y_{i+1}^2 - 2y_{i+1} + 1) - 2r^2$$

$$p_{i+1} = p_i - y_i^2 - (y_i - 1)^2 + 2(2x_i + 3) + 2y_{i+1}^2 - 2y_{i+1} + 1$$

$$p_{i+1} = p_i + 4x_i + 6 - y_i^2 - y_i^2 + 2y_i - 1 + 2y_{i+1}^2 - 2y_{i+1} + 1$$

$$p_{i+1} = p_i + 4x_i + 6 + 2(y_{i+1}^2 - y_i^2) - 2(y_{i+1} - y_i)$$

Sami Gazzah 2009

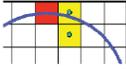
234

## I. Segments 2D, cercles

### 3. Génération de cercles

#### Algorithme de Bresenham

$$p_{i+1} = p_i + 4x_i + 6 + 2(y_{i+1}^2 - y_i^2) - 2(y_{i+1} - y_i)$$



■ On considère 2 cas :

• 1er cas :  $y_{i+1} = y_i$

$$\begin{aligned} [x_{i+1} = x_i + 1] \text{ et } [y_{i+1} = y_i] \\ \Rightarrow p_{i+1} = p_i + 4x_i + 6 \end{aligned}$$

• 2ème cas :  $y_{i+1} = y_i - 1$

$$\begin{aligned} [x_{i+1} = x_i + 1] \text{ et } [y_{i+1} = y_i - 1] \\ \Rightarrow p_{i+1} = p_i + 4(x_i - y_i) + 10 \end{aligned}$$

■ Valeur initiale:

• On trouve  $p_0 = 3 - 2r$  en résolvant l'équation de  $p_i$  avec  $(x_0, y_0) = (0, r)$

$$p_0 = (d_1 - d_2) = 2(x_0 + 1)^2 + y_0^2 + (y_0 - 1)^2 - 2r^2$$

$$= 2(0 + 1)^2 + r^2 + (r - 1)^2 - 2r^2 = 2 + r^2 + r^2 - 2r + 1 - 2r^2 = 3 - 2r$$

## I. Segments 2D, cercles

### 3. Génération de cercles

$x = 0;$

$y = r;$

$p = 3 - 2r;$

$allume(x, y);$

Boucle tant que  $x < y;$

si  $p < 0$  alors  $p = p + 4x + 6;$

sinon  $p = p + 4(x - y) + 10;$

$y = y - 1;$

fin - si

$x = x + 1;$

$allume(x, y);$

fin - boucle

$$p = d_1 - d_2 < 0 \Rightarrow y_{i+1} = y_i$$

$$p = d_1 - d_2 > 0 \Rightarrow y_{i+1} = y_i - 1$$

Sami Gazzah 2009

236

## I. Segments 2D, cercles

### 3. Génération de cercles

#### Algorithme de Bresenham :

Exemple: centre = (0,0) et  $r = 3$ .

$x_0 = 0$

$y_0 = 3$

$P_0 = 3 - 2 \times r = 3 - 6 = -3$

$x_1 = 1$

$y_1 = 3$

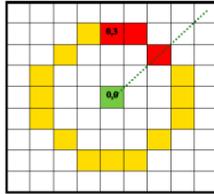
$P_1 = P_0 + 4 \times x_0 + 6 = -3 + 0 + 6 = 3$

$x_2 = 2$

$y_2 = 2$

$P_2 = P_1 + 4 \times (x_1 - y_1) + 10 = 3 - 8 + 10 = 5$

Stop (car la limite de l'octant est atteinte)



Sami Gazzah 2009

237

## Références

### Livres

- ▶ Diane Lingrand, Introduction au Traitement d'Images, Paris, Editions Vuibert, 2<sup>e</sup> édition 2008.
- ▶ Steven K. Feiner, James D. Foley, John F. Hughes, Richard I. Phillips, Andries Vanam, Introduction à l'infographie, Paris, Editions Vuibert, 2000.
- ▶ William K. Pratt, Digital Image Processing, Wiley-Inter-sciences, 4<sup>th</sup> edition, 2007.
- ▶ Éric Incerti, Compression d'image – Algorithmes et standards, Vuibert 2003
- ▶ Gilles Burel, Introduction au traitement d'images – Simulation sous Matlab, Hermès 2001.

### Sites web

- ▶ Cours de C. Rougier (U. Montréal)  
<http://www-etud.iro.umontreal.ca/~rougier/ift2730/cours.html> (cf. chapitre 3)
- ▶ Page wikipédia sur JPEG (en anglais, plus complète que celle en français)  
<http://en.wikipedia.org/wiki/JPEG>
- ▶ Cours de P. Courtellement (Campus numérique Esp@don)  
[http://perso.univ-lr.fr/~pcourtel/espadon/site\\_web/](http://perso.univ-lr.fr/~pcourtel/espadon/site_web/)
- ▶ Applet de démo, par C. G. Jennings  
<http://cgjennings.ca/toybox/hjpeg/index.html>
- ▶ Cours de D. Marshall (U. Cardiff)  
<http://www.cs.cf.ac.uk/Dave/Multimedia/PDF/>

Sami Gazzah 2009

238



The Industry's Foundation for  
High Performance Graphics

## Chapitre 9:

# introduction au traitement d'images avec OpenGL

Sami Gazzah 2009

239

## Présentation générale

- OpenGL est une librairie graphique. Elle a été créée par Silicon Graphics et bénéficie sur ces machines de l'accélération matérielle.
- Elle est constituée d'environ 200 fonctions graphiques.
- OpenGL est une bibliothèque de fonctions.
- Les fonctions sont de bas niveaux.
- La version actuelle est l'« OpenGL 4.1 » (décembre 2010)
- Principale caractéristique : indépendant du matériel et du système d'exploitation.
- Une implémentation libre de droits d'OpenGL très performante a été écrite par Brian Paul : <http://www.mesa3d.org>. Et tout son code source est accessible !

Sami Gazzah 2009

240

**OpenGL**

- OpenGL propose des fonctions de bas niveau:
  - Primitives géométriques : points, segments, polygones.
- Les fonctions de haut niveau sont développées par l'utilisateur ou importées par la librairie en amont d'OpenGL
  - Primitives géométriques complexes : Sphère, splines, etc.
- Toutes les commandes et fonctions utilisent le préfixe : gl
  - Exemple : glColor2f()

Sami Gazzah 2009 241

**OpenGL**

### Les Bibliothèques co-existant avec OpenGL

- GLU : OpenGL Utility Library contient les routines de bas niveau pour gérer les matrices de transformation et de projection, la facetisation des polygones et le rendu de surface. Les bibliothèques d'extension du système de fenêtres permettent l'utilisation du rendu OpenGL. Il s'agit de GLX pour XWindows (fonctions ayant pour préfixe glX) et de WGL pour Microsoft Windows (fonctions ayant pour préfixe wgl).
- GLUT : OpenGL Utility Toolkit est une boîte à outils indépendante du système de fenêtrage, écrite par Mark Kilgard pour simplifier la tâche d'utiliser des systèmes différents (fonctions ayant pour préfixe glut).

Sami Gazzah 2009 242

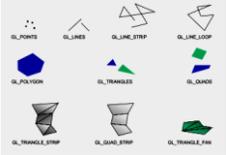
**OpenGL**

### Les primitives géométriques d'OpenGL

Les primitives sont décrites par un bloc d'instructions délimité par glBegin et glEnd

Ce bloc contient généralement une liste de vertex Exemple :

```
glBegin(GL_LINE_LOOP);
glVertex2f(-0.5,-0.5);
glVertex2f(0.5,-0.5);
glVertex2f(0.5,0.5);
glVertex2f(-0.5,0.5);
glEnd();
```



D'autres attributs peuvent être spécifiés en chaque sommet :  
couleur, normale, coordonnée de texture, etc.

Sami Gazzah 2009 243

**OpenGL**

### OpenGL est une machine à états

En modifiant les variables d'états, on peut changer le style de rendu



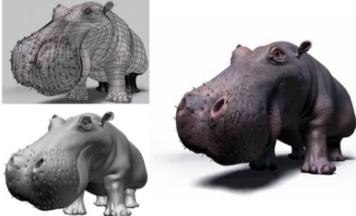
Sami Gazzah 2009 244

**OpenGL**

### Placage de textures

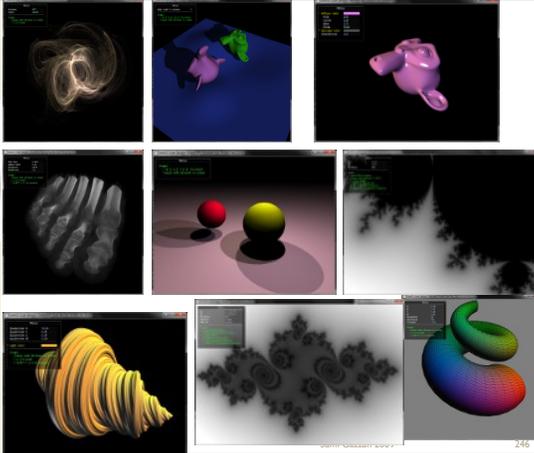
- L'apparence et le toucher ( feel ) d'une surface
- Une image utilisée pour définir les caractéristiques d'une surface
- Une image multi-dimensionnelle qui est plaquée sur un espace multi-dimensionnel

Exemple : Honda Hippo (3dRender.com)



Sami Gazzah 2009 245

**OpenGL**



Sami Gazzah 2009 246

Références: **openGL**




Sami Gazzah 2009 247



## Chapitre 10: introduction au traitement d'images avec OpenCV

Sami Gazzah 2009 248

**Présentation générale** SOURCEforge

Qu'est-ce ?

- Open Source Computer Vision Library
- Bibliothèque open source de traitement des images numériques
- Projet initié par Intel (en 1999)
- Téléchargeable sur <http://sourceforge.net/projects/opencvlibrary/>
- Fortement orientée temps réel
  - écrite en C et C++
  - optimisée pour les processeurs multi-cœurs
- Multi-plateformes : disponible pour Linux, Windows et Mac OS X

Que contient-elle ?

- Des fonctionnalités (structures et routines, >350 algos) de base et avancées pour bâtir rapidement des applications autour de la vision par ordinateur.
- Exemples d'utilisation : traitements de bas niveau (filtrage, ...), détection de visages, reconnaissance et suivi d'objets en mouvement, ...

Sami Gazzah 2009 249

**Fonctionnalités**

- Lecture/écriture d'images statiques et de séquences vidéo
- APIs de bas et moyen niveaux
  - Traitement des images binaires
  - Manipulations d'image (échantillonnage (sampling), région d'intérêt (ROI), redimensionnement (resizing), rotation, déformation (warping), ...)
  - Détection de contours, de lignes, de coins, ... (edge, ... detection)
  - Traitements multi-échelles (multiscale processing)
  - Transformées (Fourier, cosinus discrète, ...)
- Modules de haut niveau
  - Détection, reconnaissance et suivi (tracking) de visages
  - Flot optique (estimation du mouvement) (optical flow)
  - Calibrage (calibration) de la caméra
  - Mise en correspondance stéréoscopique (stereo matching)
- IA et apprentissage par ordinateur (machine learning)
- 3D : mise en correspondance (registration), ...

Sami Gazzah 2009 250

**Modules principaux**

**CxCore**

- Types de base pour manip. d'images (ex. struct. image, point E cxytypes.h)
- Structures dynamiques (listes, files, graphes, arbres) et persistantes (L/E)
- Manipulations de matrices, méthodes de l'algèbre linéaire et statistiques
- Dessin en surimpression (overlay) (ex. étiquetage)

**Cv**

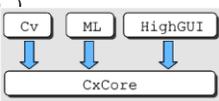
- Traitement d'image, calibrage caméra
- Géométrie algorithmique (triangulation)

**Cviaux**

- Reconnaissance de visage
- Code expérimental et obsolète

**Autres (dans otherlibs)**

- **HighGUI** : E/S vidéos, curseurs, fenêtrage
- **ML** : classification, regroupement (clustering)



Sami Gazzah 2009 251

**OpenCV : premiers programmes** Bibliothèque OpenCV

- **Chargement d'un fichier image**

```
int main(int argc, char** argv[])
{
    // charge l'image "lena.jpg" dans une structure IplImage
    IplImage* img = cvLoadImage("lena.jpg"); //ou: (argv[1])
    // crée une fenêtre nommée Exemple1, dont la taille s'adapte à son contenu
    cvNamedWindow("Exemple1", CV_WINDOW_AUTOSIZE);
    // affiche l'image "lena" (dans la structure IplImage) dans la fenêtre Exemple1
    cvShowImage("Exemple1", img);
    // attend que l'utilisateur appuie sur une touche
    cvWaitKey(0);
    // libère la mémoire image et détruit la fenêtre
    cvReleaseImage(&img);
    cvDestroyWindow("Exemple1");
    return(0);
}
```



Sami Gazzah 2009 252

**Références: openCV****Livre**

G. Bradski & A. Kaehler, « Learning OpenCV », O'Reilly Press (2008)

codes sources : <http://examples.oreilly.com/9780596516130/>

**Ressources en ligne**

Référence complète <http://opencv.willowgarage.com/documentation>

Wiki <http://opencv.willowgarage.com/wiki/FullOpenCVWiki>

- Présentation, téléchargement, installation

- avec Code::Blocks : <http://opencv.willowgarage.com/wiki/CodeBlocks>

- Débuter avec OpenCV, FAQ

**Tutoriels**

<http://nchc.dl.sourceforge.net/sourceforge/opencvlibrary/ippocv.pdf>

<http://dasl.mem.drexel.edu/~noahKuntz/openCVTut1.html> à ...Tut1 I.html

[http://www.cognotics.com/opencv/servo\\_2007\\_series](http://www.cognotics.com/opencv/servo_2007_series)

**Tutoriel sur la vision utilisant OpenCV (en français)**

<http://www.siteduzero.com/tutoriel-3-8631-vision-par-ordinateur.html>

253