

VISION PAR ORDINATEUR

IG. 2405

Florence ROSSANT

- **Introduction, codage des images**
- **Principaux traitements**
 - Traitement sur histogrammes
 - Transformations géométriques
 - Filtres
 - Descripteurs
- **Segmentation**
- **Morphologie mathématique**
- **Reconnaissance de formes (méthodes par apprentissage, ML, DL)**

VISION PAR ORDINATEUR

INTRODUCTION

Image numérique- Objectifs

Florence ROSSANT

1. Image numérique

- Acquisition
- Représentation
- Codage de la couleur
- Topologie

2. Stockage des images

3. Traitement et analyse d'images

- Définitions
- Exemple (OCR)
- Applications

4. Objectifs du module IG. 2405

5. Méthodes de travail personnel

6. Evaluation

1. Image numérique

- Acquisition
- Représentation
- Codage de la couleur
- Topologie

2. Stockage des images

3. Traitement et analyse d'images

- Définitions
- Exemple (OCR)
- Applications

4. Objectifs du module IG. 2405

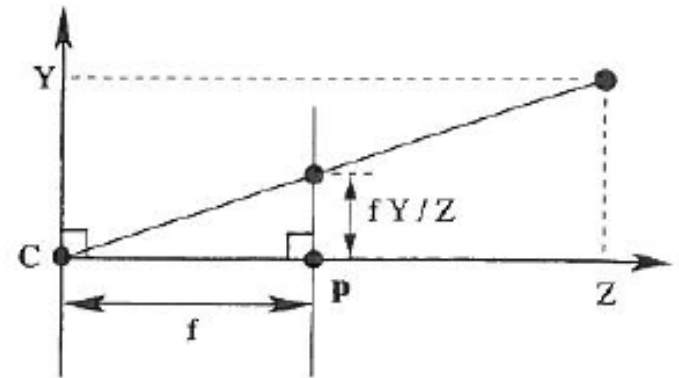
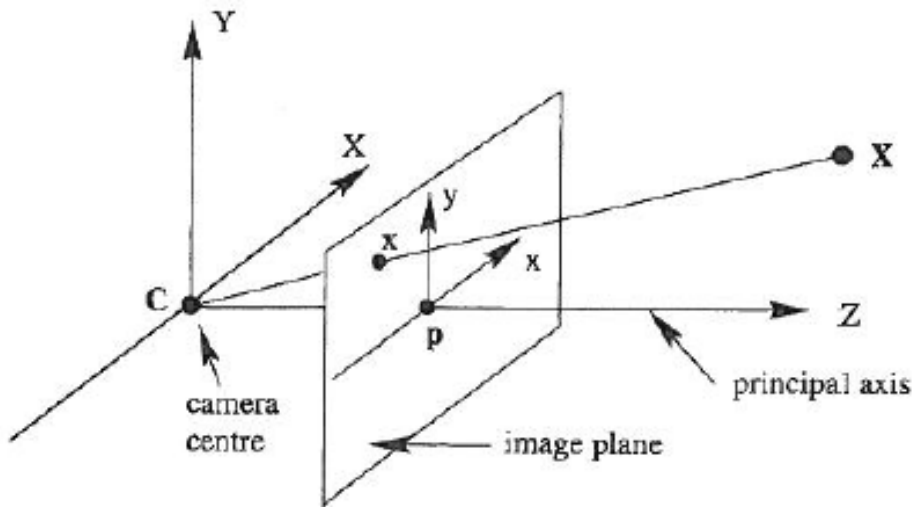
5. Méthodes de travail personnel

6. Evaluation

1. Image numérique - *Acquisition*

■ Acquisition : géométrie

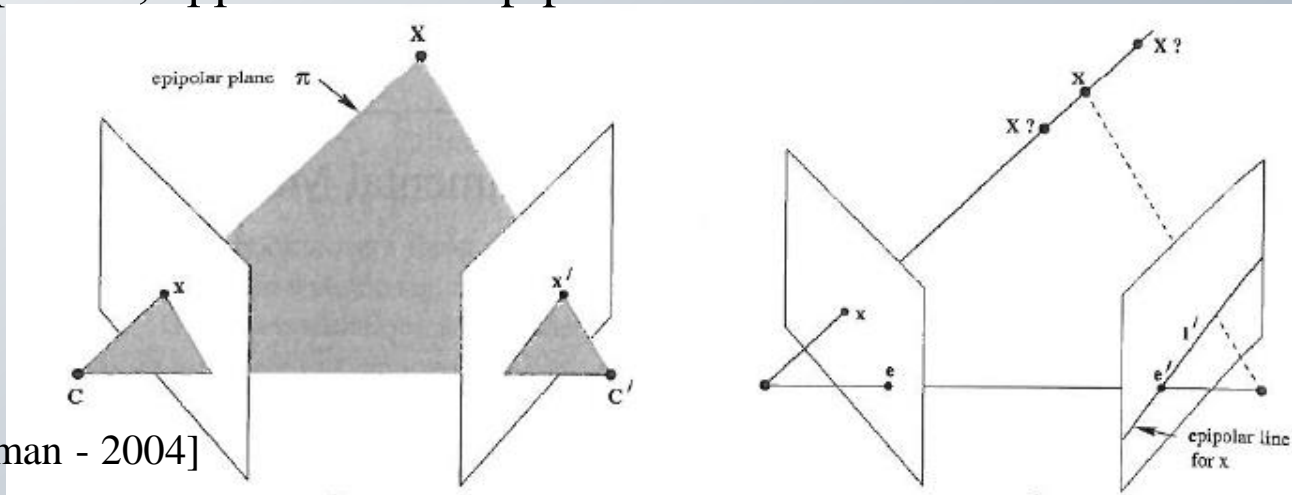
- Représentation 2D du monde réel 3D
- Obtenue par projection
- Perte d'une dimension



[Hartley, Zisserman - 2004]

■ Reconstruction 3D: géométrie épipolaire

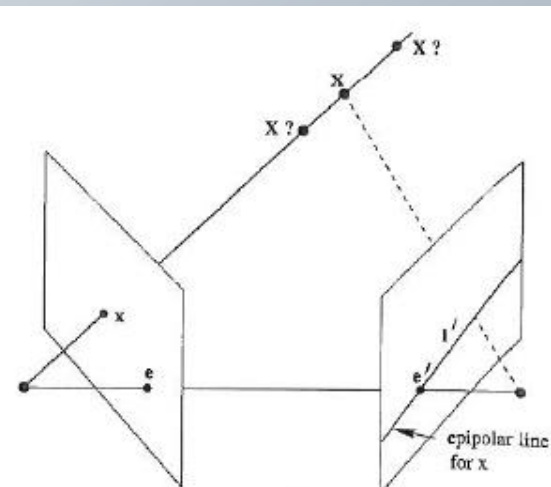
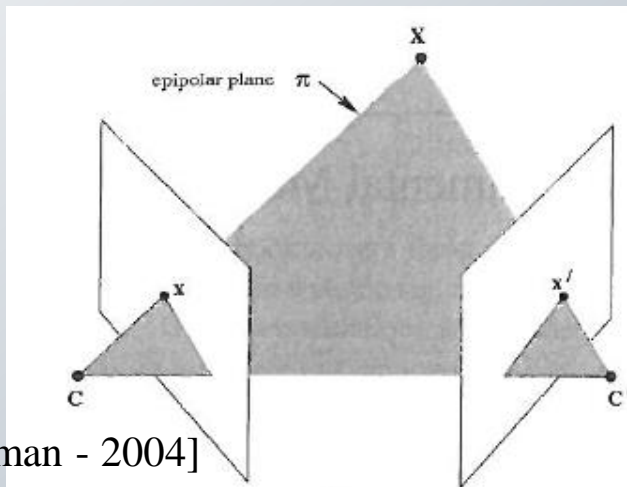
- Hyps: \mathbf{x} et \mathbf{x}' , pb: déterminer \mathbf{X} , le point de l'espace 3D qui se projette en \mathbf{x} et \mathbf{x}'
- 1. \mathbf{X} se situe sur la droite $(\mathbf{c}\mathbf{x})$: rétroprojection du point image \mathbf{x} .
- 2. \mathbf{X} , \mathbf{x} , \mathbf{x}' se situent dans le plan épipolaire contenant les centres des caméra.
- 3. Pour déterminer la position exacte de \mathbf{X} , il faut déterminer \mathbf{x} et \mathbf{x}' . \mathbf{X} se situe alors à l'intersection des deux rayons rétroprojetés.
- 4. La mise en correspondance de \mathbf{x} et \mathbf{x}' est délicate. On raisonne en géométrie épipolaire: on sait que \mathbf{x}' se situe sur l'image du rayon $(\mathbf{c}\mathbf{x})$, notée l' , contenant l'épipoles e' , appelée droite épipolaire.



[Hartley, Zisserman - 2004]

1. Image numérique - *Acquisition*

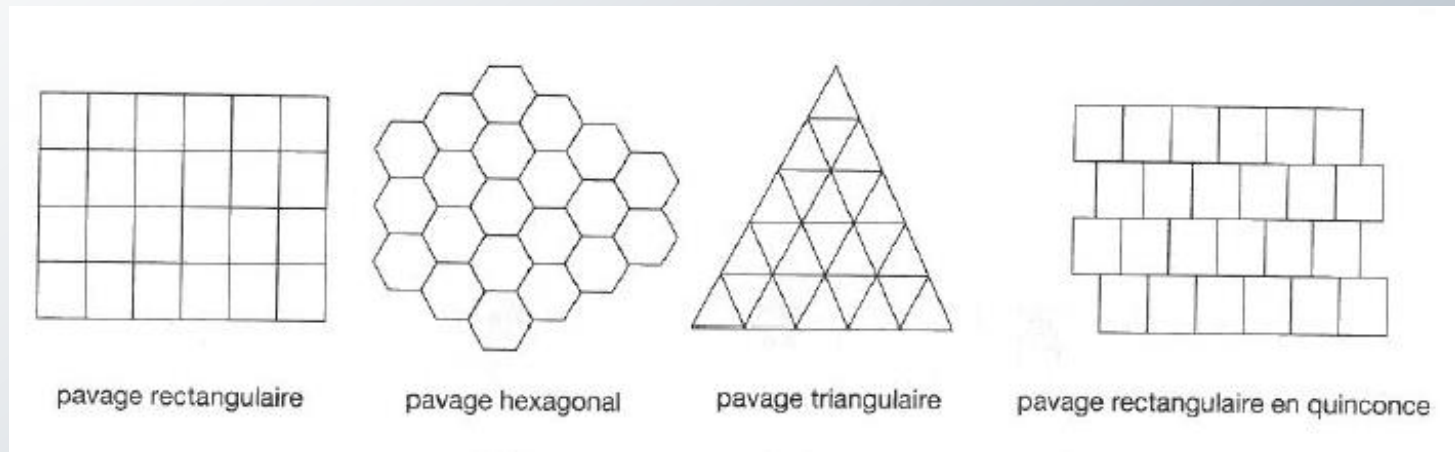
■ Acquisition : géométrie épipolaire



[Hartley, Zisserman - 2004]

■ Acquisition : discrétisation spatiale

- Partition du plan en cellules élémentaires (en lien avec les capteurs)
- Discrétisation du domaine spatial : pavage, tessellation



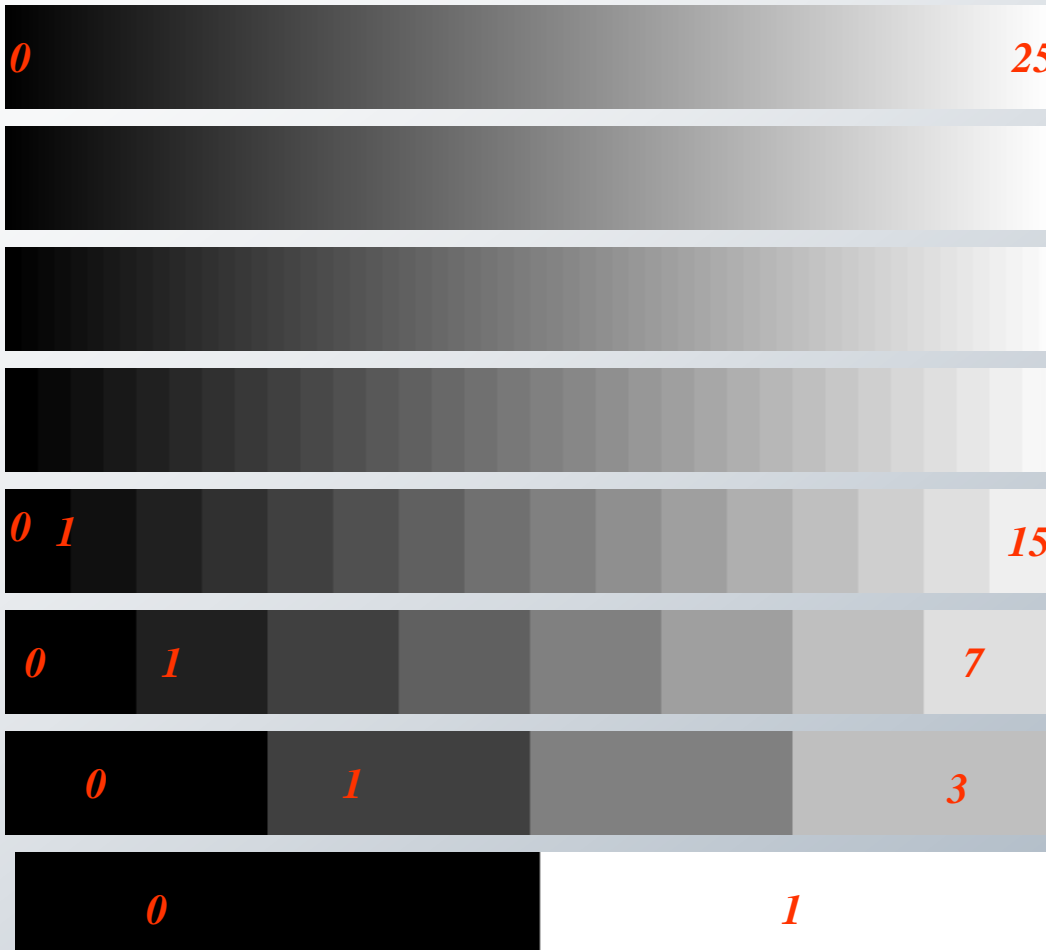
[Lingrand - 2008]

- Le plus souvent, pavage carré $\Delta x = \Delta y$
- Représentation matricielle : $f(x,y)$, avec $0 \leq x < H$, $0 \leq y < W$
- $f(x,y)$ représente la couleur, le niveau de gris, du pixel de coordonnées (x,y)

Acquisition : quantification

Signal continu

- Signal quantifié sur b bits, codé par N entiers



$b = 8, N =$

$b = 7, N =$

$b = 6, N =$

$b = 5, N =$

$b = 4, N =$

$b = 3, N =$

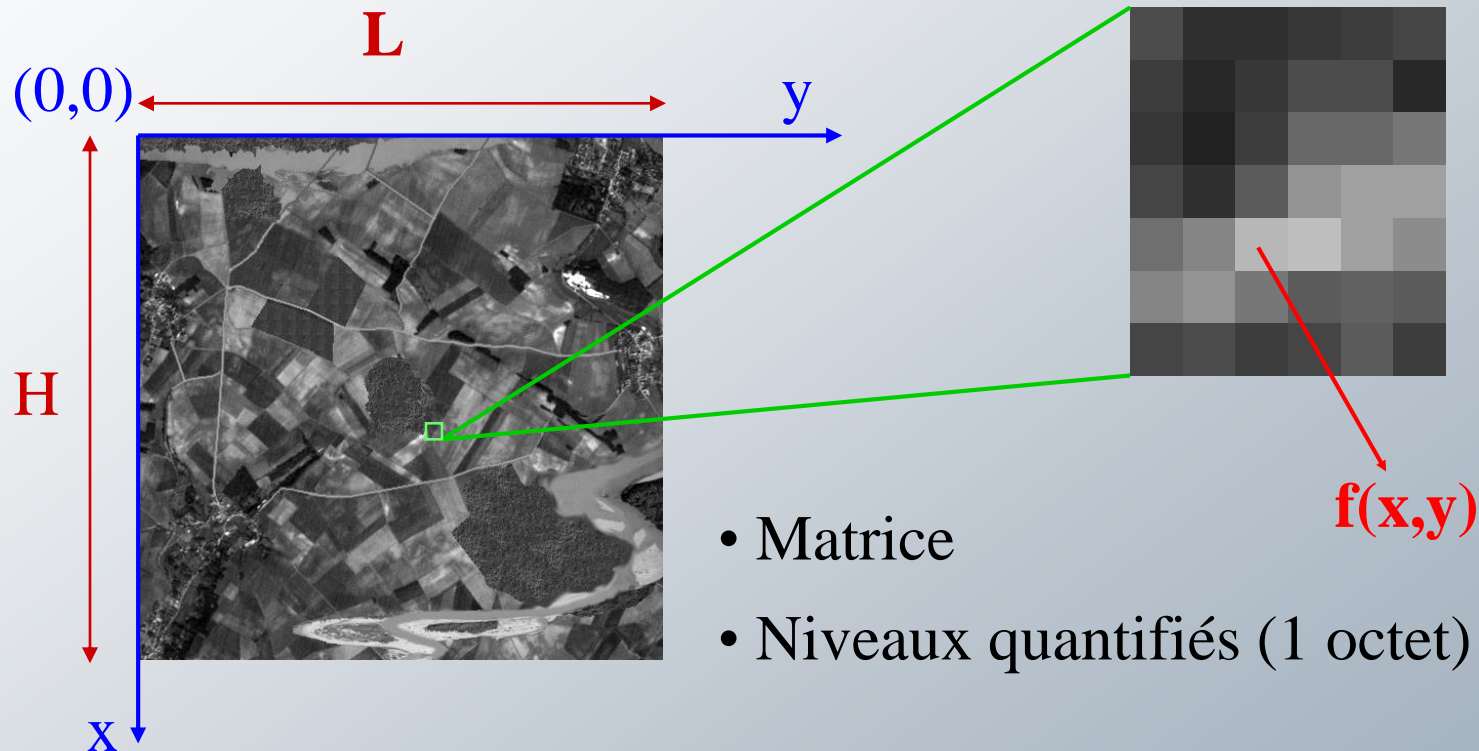
$b = 2, N =$

$b = 1, N =$

1. Image numérique - Représentation

■ Représentation d'une image fixe 2D en niveaux de gris

- Pavage carré
- Quantification sur 8 bits



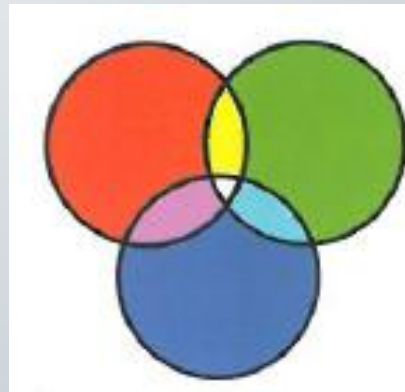
- Matrice
- Niveaux quantifiés (1 octet)

$$f(x, y) = i \quad \text{avec } i \in [0, 255]$$

$$0 \leq x < H, \quad 0 \leq y < W$$

■ Représentation d'une image fixe 2D couleur

- Trichromie: une couleur peut-être synthétisée à partir de 3 primaires
- Synthèse additive: les couleurs sont synthétisées par la somme pondérée des trois primaires R, V et B, respectivement le rouge ($\lambda=700$ nm), le vert ($\lambda=546$ nm), le bleu ($\lambda=435.8$ nm)



[Lingrand - 2008]

- Il existe d'autres espaces de couleur, CMY (Cyan, Magenta, Yellow, utilisé en imprimerie), HSV (Hue, Saturation, Value)

1. Image numérique - *Couleur*

■ Représentation d'une image fixe 2D couleur

- Généralement, chaque canal est codé sur 1 octet.
- La couleur d'un pixel est donc codée par 3 octets.
- Le nombre de couleurs possibles est donc : _____

(255,0,0)	(255,255,0)	(0,255,0)	(0,255,255)	(0,0,255)
(255,0,255)	(0,0,0)	(63,63,63)	(128,128,128)	(255,255,255)

■ Passage couleur → luminance

- Pixel couleur : (R, G, B)
- Luminance :

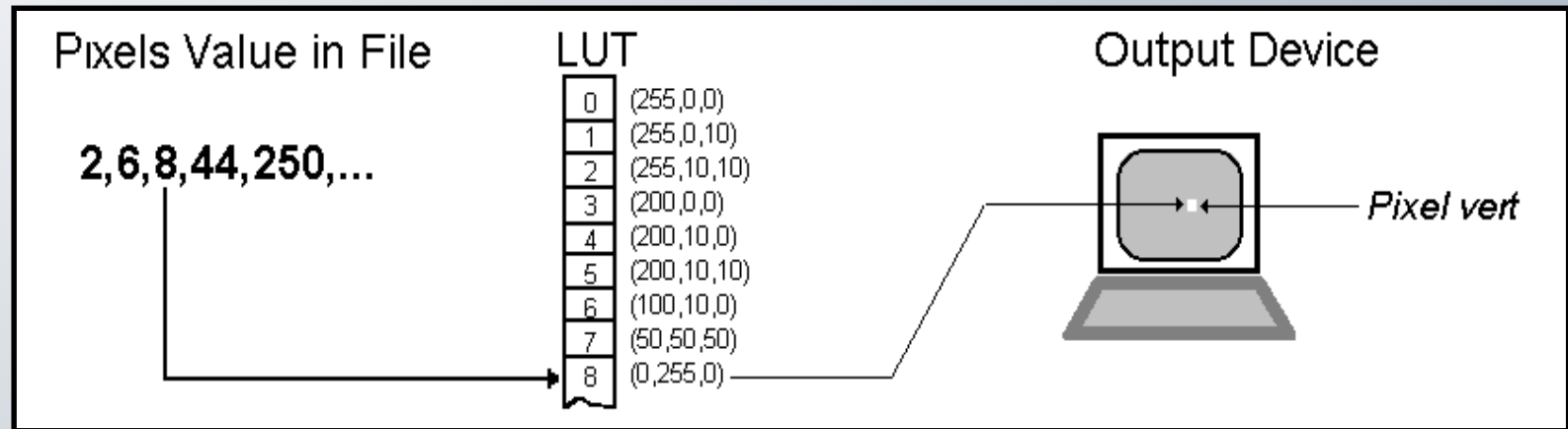
$$Y = 0.299R + 0.587G + 0.114B$$



1. Image numérique - *Couleur*

■ Table des couleurs (Look Up Table, LUT)

- Dans certains cas, on représente l'ensemble des couleurs disponibles par un tableau, la table des couleurs.
- Chaque entrée, d'indice k , de la table des couleurs est un triplet (R, G, B) qui représente une couleur
- La couleur de chaque pixel est codée par un indice, entrée de la table



■ Table des couleurs (Look Up Table, LUT)

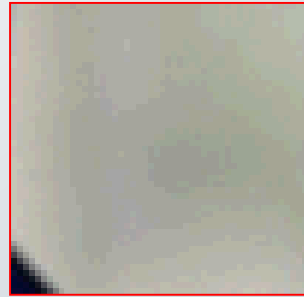
- Dans certains cas, on représente l'ensemble des couleurs disponibles par un tableau, la table des couleurs.
 - Chaque entrée, d'indice k , de la table des couleurs est un triplet (R, G, B) qui représente une couleur
 - La couleur de chaque pixel est codée par un indice, entrée de la table
 - Intérêt:
 - ✓ Gain de place si le nombre de couleurs de l'image n'est pas trop grand
 - ✓ Possibilité de modifier l'affichage sans modifier les pixels eux-mêmes
- Exemple: affichage en fausses couleurs

1. Image numérique - *Couleur*

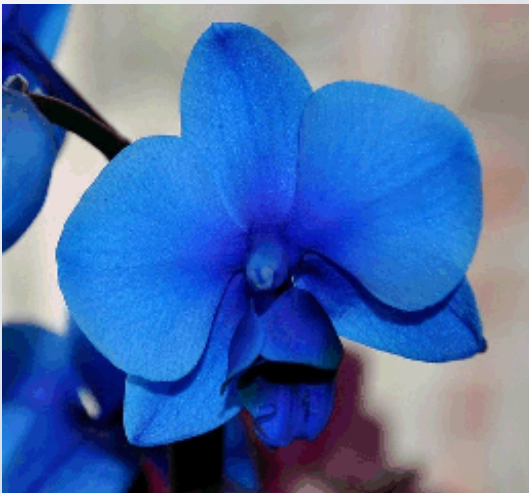
■ Définition des couleurs

- Image 240 x 250 pixels

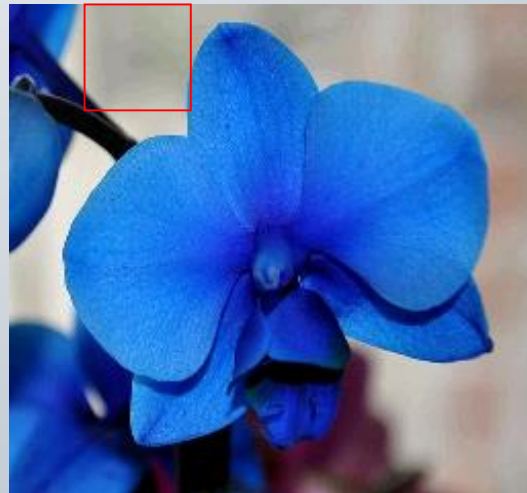
- Image codée en RGB,
1 octet par canal : _____



RGB, 16 millions de couleurs



256 couleurs



32 couleurs



16 couleurs

■ Définition des couleurs

- L'image est codée sur 256 couleurs, 1 octet par canal
- Les 25 premières couleurs de la table des couleurs sont représentées ci-contre (indice de la première ligne : 0)
- Combien d'octets sont nécessaires pour coder un pixel ?
- Indiquer la couleur des pixels codés par

✓ 0 :

✓ 2 :

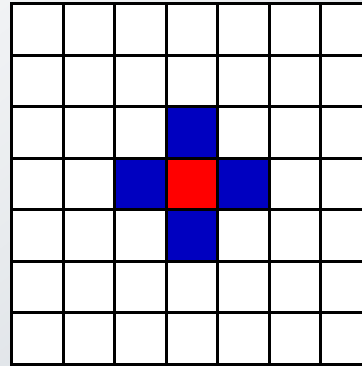
✓ 7 :

✓ 10 :

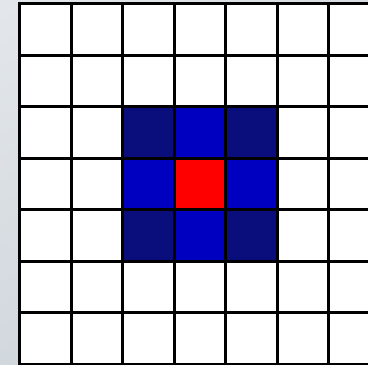
0	0	0
0	8	0
0	0	16
0	11	14
8	2	6
7	14	17
24	6	12
26	21	21
0	0	41
14	6	39
0	0	69
15	4	58
15	16	46
13	12	71
17	32	53
10	33	75
0	6	93
0	21	96
8	8	88
19	20	85
0	8	111
14	16	103
1	8	131
15	16	124
0	28	98

■ Topologie

- Notion de voisinage



Connexité 4



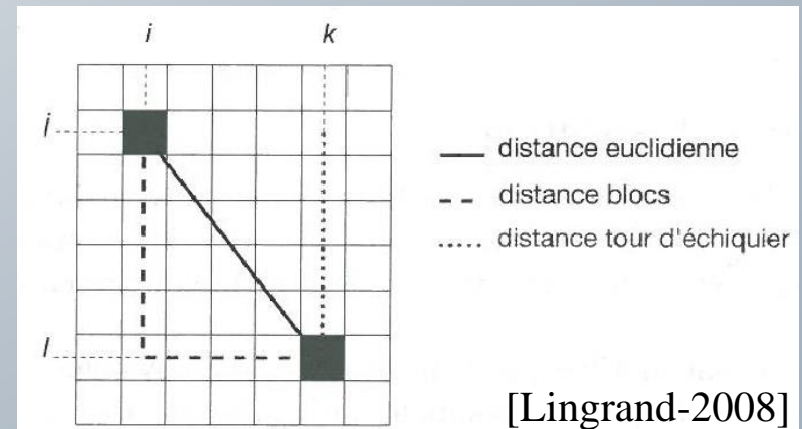
Connexité 8

- Mesures de distances : euclidienne, city-blocks, tour d'échiquier

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

$$d_{blocks}(P(i, j), P(k, l)) = |i - k| + |j - l|$$

$$d_{ech}(P(i, j), P(k, l)) = \max(|i - k|, |j - l|)$$



[Lingrand-2008]

1. Image numérique

- Acquisition
- Représentation
- Codage de la couleur
- Topologie

2. **Stockage des images**

3. Traitement et analyse d'images

- Définitions
- Exemple (OCR)
- Applications

4. Objectifs du module IG. 2405

5. Méthodes de travail personnel

6. Evaluation

■ Stockage des images : fichiers graphiques

- De nombreux formats de fichiers graphiques : PGM, BMP, TIFF, GIF, JPEG,...
- Entête:
 - Un certain nombre de champs fixes, non compressée
 - Informations permettant la compréhension du fichier
 - Identificateur de fichier, numéro de version.
 - Taille de l'image, profondeur.
 - type de compression.
 - présence d'une palette.
 - ...
- Table des couleurs
- Données
 - ✓ Non compressées
 - ✓ Compression RLE, LZW, JPEG

Entête
Table des couleurs
Data
Footer

■ Stockage des images : fichiers graphiques

- Exemple: format PGM



Format PGM – données ASCII

Commentaire

Nb colonnes Nb lignes

Valeur pixel max

Données : balayage ligne

```
P2
# Created by Paint Shop Pro 5
429 213
255
95 93 92 97 96 95 94 93 94 94 95 98 99 99 99
102 107 110 109 107 107 108 108 109 109 110
117 119 116 118 120 121 121 121 122 124 123
132 132 131 130 130 131 132 132 130 131 131
128 127 126 126 127 127 125 126 128 128 127
130 130 130 130 132 130 128 128 130 131 131
135 134 133 133 135 135 136 135 135 134 137
```


1. Image numérique

- Acquisition
- Représentation
- Codage de la couleur
- Topologie

2. Stockage des images

3. **Traitement et analyse d'images**

- Définitions
- Exemple (OCR)
- Applications

4. Objectifs du module IG. 2405

5. Méthodes de travail personnel

6. Evaluation

- **Traitement : opération qui transforme une image en une autre image:**

$$f'(x_0, y_0) = T \left(f(x, y)_{\substack{0 \leq x < L \\ 0 \leq y < H}} \right) \quad \text{pour} \quad \begin{cases} 0 \leq x_0 < L' \\ 0 \leq y_0 < H' \end{cases}$$

- Exemple : réduction de taille d'un facteur 2 par moyennage

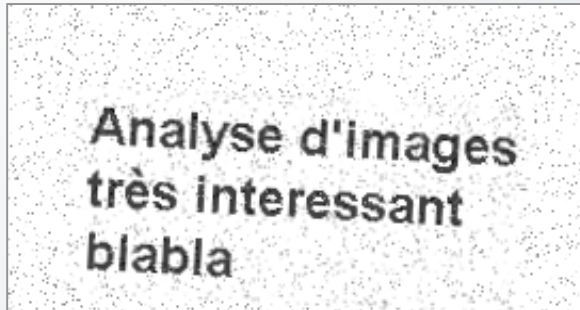
$$f'(x_0, y_0) = \frac{1}{4} \sum_{i=0}^1 \sum_{j=0}^1 f(2 * x_0 + i, 2 * y_0 + j) \quad \text{pour} \quad \begin{cases} 0 \leq x_0 < L/2 \\ 0 \leq y_0 < H/2 \end{cases}$$

- Exemples:
 - ✓ Transformations géométriques
 - ✓ Amélioration de la qualité des images (rehaussement, restauration)

- **Analyse : extraction d'informations (vision artificielle)**
 - Reconnaître un objet
 - Mesurer la taille d'un objet
 - ...
- **Phase qui fait suite aux traitements**
- **Exemples**
 - Identifier une personne (biométrie)
 - Mesurer des caractéristiques morphologiques (imagerie médicale)
 - Détecter une pathologie (imagerie médicale)
 - Détecter des intrusions (vidéo surveillance)
 - ...

■ Un exemple : OCR

Acquisition

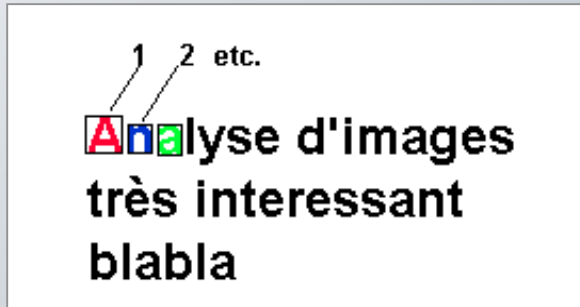


- Numérisation

Traitements bas niveaux



- Transformation géométrique
- Débruitage
- Binarisation



- Segmentation :
 - Partition de l'image en ensembles de pixels connexes :
1 ensemble \leftrightarrow 1 objet ayant un « sens »

3. Traitement et Analyse- *Exemple*

Un exemple : OCR



- Description symbolique (modélisation)
 - Positions
 - Identification des segments, des arcs constituant les lettres
 - Extraction d'attributs caractérisant les objets :
 - Points triples, h/l,
 - nombre de segments horizontaux
 - etc.
- Classification
 - $S_1=A, S_2=n, S_3=a, S_4=l, S_5=g, S_6=s, \text{ etc.}$
- Analyse sémantique
 - $M_1=\text{Analyse} \rightarrow \text{Analyse}$

Reconnaissance (classification)

Post-
traitements

- Restauration des images spatiales (turbulences atmosphériques, flou ...)
- Robotique : « vision » des robots
- Aide au pilotage automatique de véhicules
- Contrôle de qualité sur chaînes de production
- Authentification :
 - signatures
 - empruntes digitales, iris de l'oeil
- Lecture automatique : adresse (poste), montant de chèques, texte scanné...
- Interprétation d'images dans le domaine médical (différentes sources -> fusion)
- Détection d'intrusion dans des lieux non gardés
- Reconstitution 3D stéréoscopie
- Télédétection, cartographie (fusion d'images)
- Compression des images (multimédia)

1. Image numérique

- Acquisition
- Représentation
- Codage de la couleur
- Topologie

2. Stockage des images

3. Traitement et analyse d'images

- Définitions
- Exemple (OCR)
- Applications

4. Objectifs du module IG. 2405

5. Méthodes de travail personnel

6. Evaluation

- **Traitement et analyse d'images « fixes » en niveaux de gris**
- **Cours orienté « vision par ordinateur »**
 - Traitements de « bas niveaux » (i.e. transformation d'images)
 - Segmentation
 - Modélisation de « haut niveau » (e.g. descripteurs)
 - Reconnaissance (détection, classification)
- **De nombreux algorithmes pour chaque étape...**
 - Des algorithmes « classiques » de traitement et d'analyse d'images
 - Des méthodes par apprentissage: machine learning et deep learning
- **On ne verra pas :**
 - La compression
 - L'analyse de flux vidéo
 - La stéréoscopie

■ Compétences

- Acquérir les bases du traitement et de l'analyse des images numériques, i.e. les principaux algorithmes utilisés dans les systèmes de vision artificielle
- Mettre en œuvre ces algorithmes
- Identifier les principales étapes d'un système permettant de résoudre un problème de vision par ordinateur
- Choisir la meilleure approche en fonction du problème posé et des contraintes (e.g. coût de calcul, coût « humain », performances attendues, disponibilité de données annotées ou non, etc.)
- Pour chaque étape, choix des algorithmes en fonction du problème posé
 - Implémentation
 - Test
 - Evaluation (vérité terrain, métriques)
- ✓ Savoir traduire un concept mathématique en algorithme
- ✓ Evaluation quantitative des résultats obtenus par la chaîne proposée

■ Cours :

- Présentation des concepts et des algorithmes.
- Illustrations, champ d'application

■ Tds

- Mise en pratique sur des exemples simples, en Python
 - ✓ Savoir traduire les concepts mathématiques en algorithmes et les coder
 - ✓ Comprendre en pratique l'intérêt de chaque algorithme, son champ d'application, et savoir en évaluer la complexité algorithmique.
 - ✓ Mise en œuvre sur des applications de VpO

■ Projet

- Mise en application sur un sujet donné

■ Travail personnel

- **Participer et poser des questions**
- **1 heure de cours/td => 1 heure de travail personnel**
 - Compréhension des cours: se poser quelques questions...
 - ✓ Ai-je compris le champ d'application de la méthode présentée ?
 - ✓ Suis-je capable d'appliquer la méthode ?
 - ✓ Suis-je capable de passer de la formalisation mathématique (équations) à l'implémentation algorithmique ?
 - Revoir les Tds en autonomie, faire les exercices de travail personnel proposés dans les Tds.
 - Compléter si besoin par d'autres sources
 - Implémenter et expérimenter

■ **Contrôle continu : 14 points**

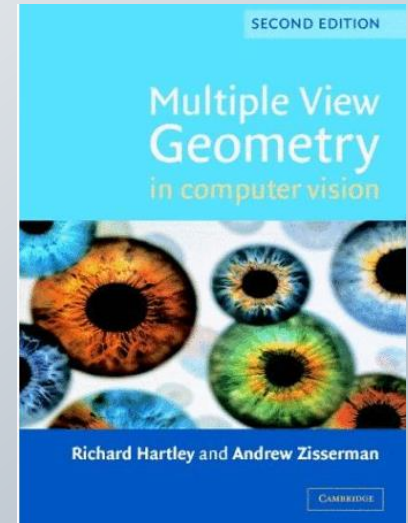
- Examen d'une heure tous les 3 / 4 cours environ : applications directes et « basiques » des méthodes présentées (C4 – C7 - C11 - C14)

■ **Projet : 6 points**

- Etude bibliographique
- Conception d'un système de VpO
 - ✓ Formalisation
 - ✓ Implémentation
 - ✓ Simulations
 - ✓ Evaluation quantitative
- Démarche ingénieur

- **[Hartley, Zisserman - 2004]**

Richard Hartley, Andrew Zisserman,
Multiple view geometry in computer vision,
CAMBRIDGE UNIVERSITY PRESS, second edition,
ISBN 978-0-521-54051-3



- **[Lingrand - 2008]**

Diane Lingrand,
Introduction au traitement d'images,
VUIBERT éditions,
ISBN 978-2-7117-4866-2

