

Introduction au traitement d'images

Marie-Odile Berger, INRIA Nancy Grand Est

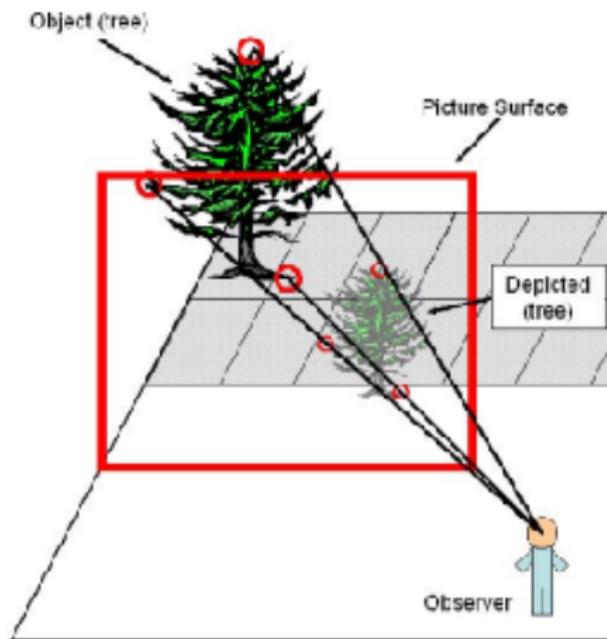
- ▶ les fondements du Traitement d'Images
 - ▶ formation d'une image, bas niveau
 - ▶ quelles informations peut on extraire d'une image 2D ?
 - ▶ reconstruire en 3D à partir d'images
- ▶ outils de test et de démonstration : matlab + toolbox image
processing, machine learning
- ▶ objectifs :
 - ▶ vous montrer qu'on peut rapidement construire des applications de vision utiles dans votre discipline
 - ▶ vous apprendre comment fonctionnent les nouvelles méthodes de [machine learning](#) et les réseau de neurones

Première partie I

Formation d'une image : différents types
d'images

Formation d'une image

- ▶ Une image est une représentation bi-dimensionnelle d'une scène.
- ▶ Elle dépend de la scène (géométrie), du système optique (lentilles, focales,...), des propriétés associées à la lumière (illumination, propriétés de réflectance des matériaux de la scène)



Formation d'une image

Les objets émettent une onde électromagnétique, ils en reflètent aussi.

- ▶ le cas le plus fréquent : les rayons lumineux issues de sources (lampes, soleil,...) sont réfléchis par les objets et atteignent le capteur.
- ▶ Les objets émettent des ondes électromagnétiques : c'est le cas de l'infra-rouge
- ▶ les objets absorbent des ondes et on mesure le reste : c'est le cas des images RX.

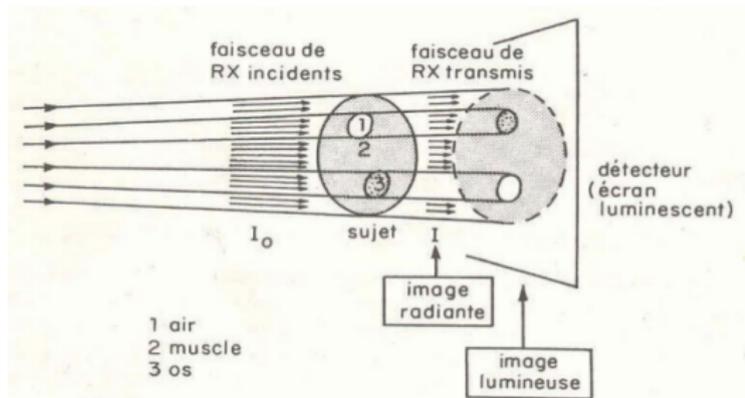
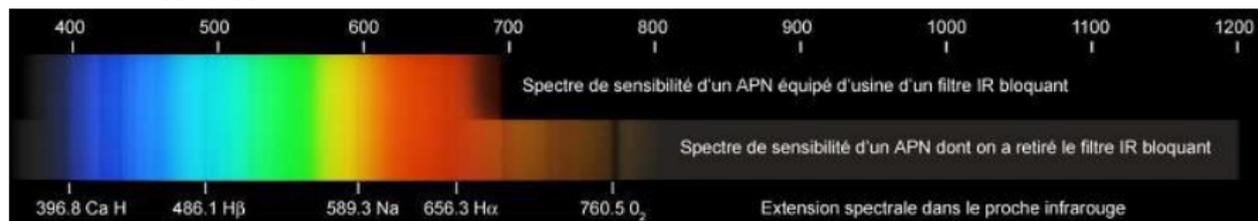


Fig. Image Radiante

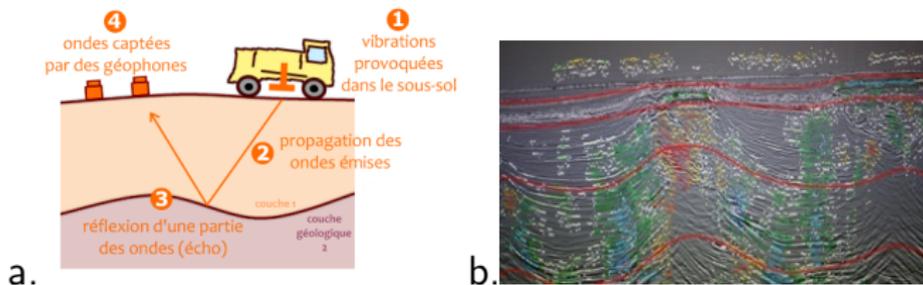
Différents types de formation d'une image

- ▶ les objets émettent une onde électromagnétique, ils en reflètent aussi.
 - ▶ Le niveau de gris d'une image est l'énergie reçue par le capteur.
 - ▶ Dans des images multi-spectrales, chaque composante mesure l'éclairement reçu dans une gamme de fréquences du signal électromagnétique (pas forcément dans le domaine visible). exemple : l'imagerie infra-rouge est utilisée pour détecter les zones de chaleur, invisibles à l'oeil nu. L'oeil humain voit les longueurs d'onde entre 400nm et 750 nm mais les capteurs sont sensibles aux ondes entre 200 et 1200 nm.
 - ▶ les images couleurs : Dans le modèle RGB, les couleurs sont obtenues par addition des trois couleurs fondamentales Rouge ($\lambda = 700nm$), vert et bleu.



Images sismiques (image d'échos)

- ▶ La sismique par réflexion étudie la réflexion d'ondes sismiques aux interfaces entre plusieurs couches géologiques.
- ▶ Les ondes émises se propagent suivant les lois de réflexion et de réfraction et sont en partie réfléchies à chaque changement de vitesse de propagation (couche géologique). Elles sont mesurées par des capteurs.
- ▶ Traitement : transformer les données sismiques en une image en deux ou trois dimensions (2D ou 3D) du sous sol



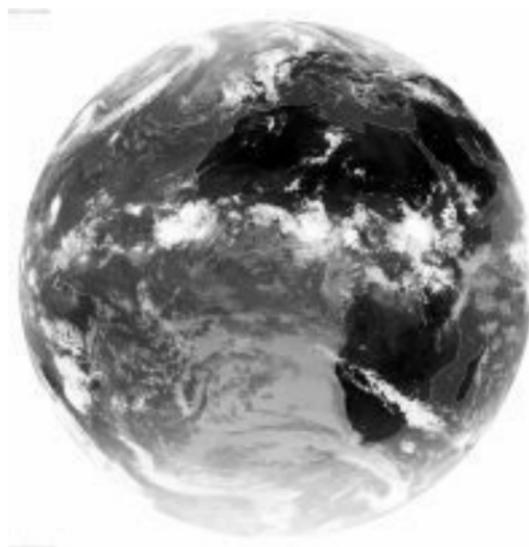
- ▶ permet par exemple d'estimer les probabilités de trouver du pétrole.

Image infrarouge meteo

Exemple tiré de <http://fr.allmetsat.com/interpretation.php>

Les images infrarouges représentent une mesure du rayonnement infrarouge émis par le sol ou les nuages. Ce rayonnement dépend de la température.

- ▶ Plus l'objet est chaud, plus il est noir et plus l'objet est froid, plus il est blanc.
- ▶ Les nuages élevés apparaissent plus blancs que les nuages bas car ils sont plus froids.
- ▶ Dans les zones sans nuages, plus le sol est chaud, plus il est sombre.



Beaucoup d'autres images

- ▶ beaucoup d'images créées par la physique-chimie
- ▶ images de profondeurs (stéréo multi-vues, laser, Kinect,...)
- ▶ les séquences d'images 2D+t (temporelles)
- ▶ images volumiques (scanner, IRM,...)
- ▶ séquences d'images volumiques
- ▶ nombreux exemples en imagerie médicale

Exemple d'images 3D



stereo multivues (56 frames)



Kinect



Laser Scanning 3D

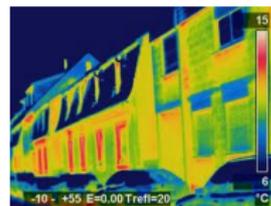
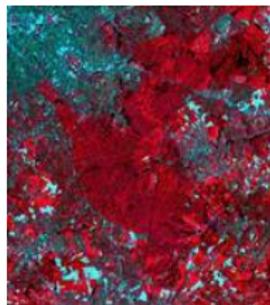


Deuxième partie II

Visualisation des images

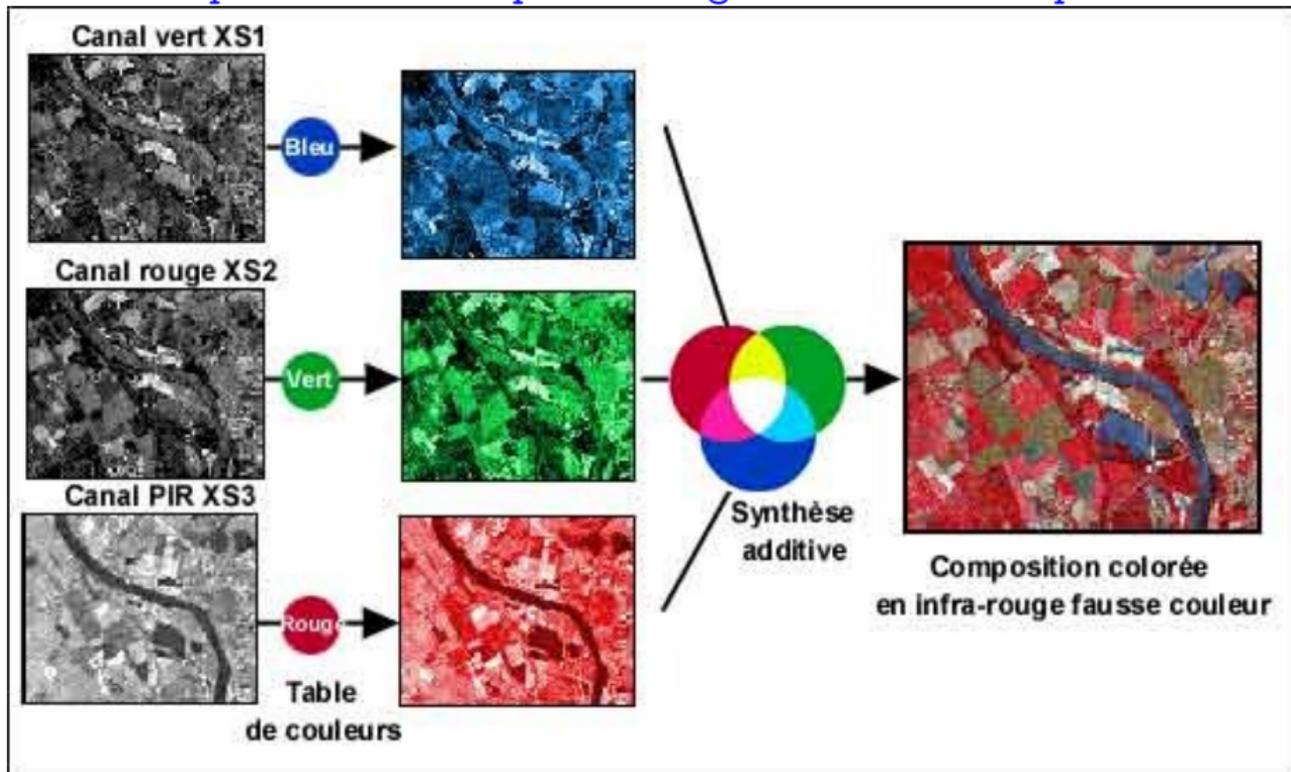
La visualisation des images

- ▶ Une image correspondant à un seul canal peut être rendue avec une couleur proportionnelle à l'intensité (en niveaux de gris)
- ▶ Une image couleur est fidèle à ce qu'un observateur aurait vu
- ▶ On peut aussi utiliser de fausses couleurs *lut* ou *colormap* pour une meilleure compréhension du rendu (sous matlab : `edit`→`colormap`).
- ▶ les images infrarouges sont souvent rendues avec une échelle de rouge
- ▶ les images d'erreurs avec une couleur d'autant plus chaude que l'erreur est importante



La visualisation des images

Tiré de <http://eoeu.belspo.be/fr/guide/deftele.asp?section=1>

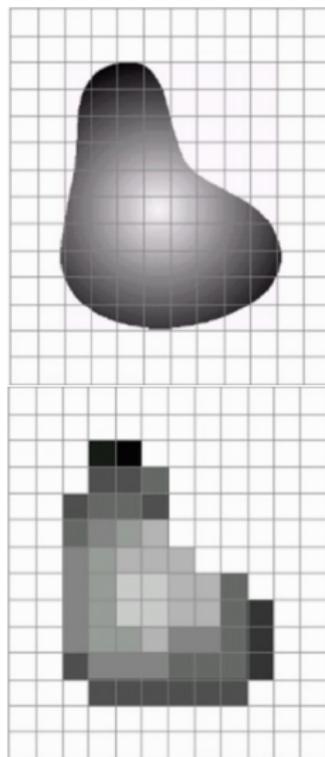


Troisième partie III

Quantification et compression d'une image

La réalité d'une image numérique

- ▶ Une image est un tableau à deux dimensions à valeurs discrètes
- ▶ L'image est issue d'un monde continu par pixelisation et quantification des niveaux de couleur
- ▶ En pratique, on considèrera une image comme une fonction $R \rightarrow R$ mais elle n'est connue qu'en un ensemble discret de valeurs.



Quantification des images

- ▶ Chaque valeur d'une composante est représentée par un mot binaire codé sur un nombre fini de bits :
 - ▶ codage 8 bits : on code $2^8 = 256$ valeurs
 - ▶ codage sur 16 bits : 65536 valeurs
 - ▶ n bits : codage entre 0 et $2^n - 1$ valeurs
- ▶ Effets de la quantification sur la qualité des images sur une image 50x50 avec 255 puis 16 et 8 niveaux de gris :



Les capacité de stockage sont maintenant très grandes mais on produit de plus en plus d'images, de séquences à visualiser sur des dispositifs portables → nécessité de compresser les données

- ▶ deux critères antagonistes : le taux de compression et la qualité de l'image après compression.
- ▶ $\text{taux compression} = \frac{\text{nombre de bits utilisés par l'image originale}}{\text{nombre de bits utilisés par l'image compactée}}$
- ▶ le codage peut être sans perte (GIF, TIFF, png) ou avec perte (JPG, JPG 2000). Quantifier la perte est difficile car c'est aussi une question de perception.

- ▶ **Pourquoi peut on compresser ?** : car les valeurs des pixels ne sont pas indépendantes mais sont corrélées à leurs voisins.
- ▶ Exemples de méthodes pour la compression **sans perte** :
 - ▶ méthode de codage des répétitions
 - ▶ méthodes statistiques : le codage de Huffman utilise un code à longueur variable pour représenter un symbole de la source. Le code est déterminé à partir des probabilités d'apparition des symboles de source : un code **court** est associé aux symboles de source les plus **fréquents**.
- ▶ Principes de compression **avec perte**
 - ▶ réduction/sous échantillonnage de l'espace des couleurs
 - ▶ approximations locales de l'image avec des transformations

JPEG= Joint Photographic Expert Group

- ▶ Taux de compression (20 :1 à 25 :1 sans perte notable de qualité)
- ▶ Découpage de l'image en blocs de 8x8 points
- ▶ puis l'application de la fonction DCT (Discrete Cosinus Transform, transformation discrète en cosinus : variante de la transformée de Fourier) qui décompose l'image en somme de fréquences.

Dans la norme JPEG2000, utilisation d'une transformée en ondelettes.

$$\text{DCT}(i, j) = \frac{2}{N} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{pixel}(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right]$$

La compression s'effectue sur les hautes fréquences : selon un seuil, on met à 0 les valeurs peu significatives et on arrondit les autres

- ▶ utilisation d'un algorithme de compression sans perte sur les coefficients en zigzag.

Les principales fonctionnalités :

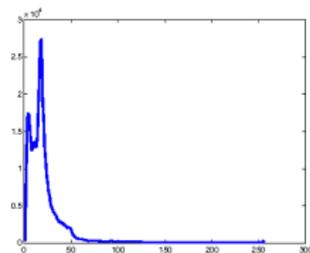
- ▶ **path** : sert à définir les répertoires où Matlab cherche par défaut les fonctions appelées ou les images ; Ajouter le répertoire ImagesTest dans votre path.
- ▶ **lire une image** : `im=imread('triGrain.jpg')` ; i est une matrice d'entiers (entre 0 et 255) de dimension 2. $im(i,j)$ est la valeur du pixel correspondant à la i^{ieme} ligne et la j^{ieme} colonne de l'image. ex : `im(100,200)`.
- ▶ connaître la **taille de l'image** : `size(im)` Les indices commencent à 1 en matlab.
- ▶ **afficher une image** : `imshow(im)`, `imagesc` (si c'est une image à valeurs non entières). Si affichage dans une **autre** fenetre : `figure` ; `imshow(im)` ;
- ▶ **extraire une sous image** `subim=im(50 :100, 100 :300)` ; `imshow(subim)` ;
- ▶ image couleur : `I=imread('champ.jpg')` ; `Taper size(I)` ; Il y a 3 composantes R,V,B. Visualiser la composante verte : `imshow(I(:, :,2))` ;
- ▶ **plot pour afficher des courbes** : `plot([1,100,200],[200,10,300])` ;
- ▶ Connaître la liste des **fonctions disponibles dans la toolbox image processing** : `help images`
- ▶ connaître la façon d'utiliser une fonction : `help nom-fonction`
- ▶ ex : avec l'image `champ.jpg` : passer d'une image couleur i à une image à niveaux de gris avec `rgb2gray`, binariser cette image avec un seuil donné avec `im2bw`

TP : Amélioration d'images et égalisation d'histogrammes

- ▶ Nécessité fréquente d'améliorer les images en augmentant le contraste (pour une meilleure visibilité ou mieux les comparer)
- ▶ Le contraste est défini comme la variance des niveaux de gris (N : nombre de pixels) :

$$1/N \sum_{x,y} (I(x,y) - moy)^2$$

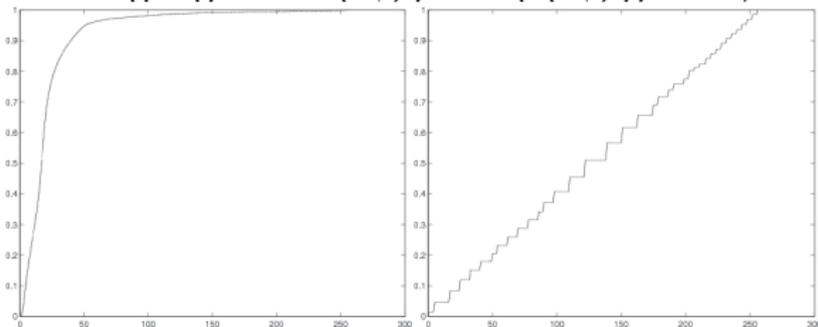
- ▶ l'égalisation d'histogramme vise à transformer l'image pour avoir une répartition uniforme des niveaux de gris



Une image avec un faible contraste

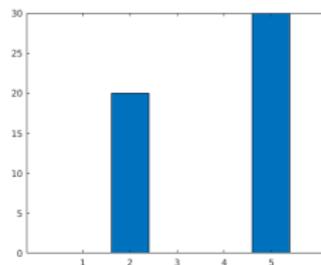
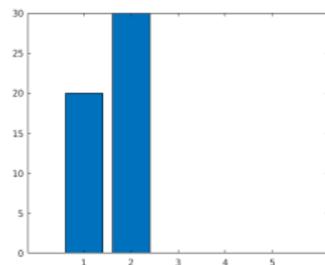
Quelques idées sur l'égalisation d'histogrammes

- ▶ objectif : avoir un histogramme le plus uniforme possible, où les niveaux de couleurs sont représentés de manière assez équitable...
- ▶ Soit N le nombre de pixels de l'image. soit hc l'histogramme cumulé de l'image. $hc(x) = \#\{p \text{ tel que } I(p) \leq x\}$
- ▶ pour une image avec une représentation uniforme : $h(\text{coul}) = N/255$
- ▶ la fonction hc devrait être une droite d'équation $hc(\text{coul}) = \text{coul} * N/255$
- ▶ pour un pixel (x,y) , la valeur $hc(I(x,y))$ devrait correspondre dans un histogramme égalisé à la couleur $hc(I(x,y))/(N/255)$
- ▶ on construit l'image égalisée $I'(x,y) = hc(I(x,y)) * 255/N$



Quelques idées sur l'égalisation d'histogrammes

exemple 1 : des couleurs proches avec des valeurs bins très élevées
5 niveaux de couleur, 50 pixels dans l'image, valeur moyenne des bins : 10
 $h(1)=20$; $h(2)=30 \rightarrow hc(1)=20$; $hc(2)=50$;
points d'intensité 1 transformés en $20/10 = 2$; points d'intensité 2 transformés en $50/10 = 5$



Quelques idées sur l'égalisation d'histogrammes

exemple 2 : des couleurs proches avec de petites valeurs

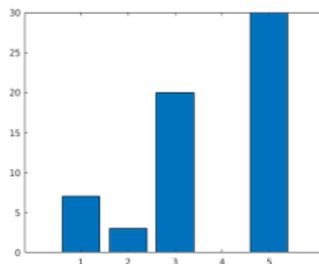
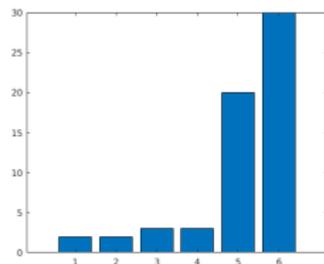
$h(0)=2, h(1)=2, h(2)=3, h(3)=3, h(4)=20, h(5)=30$

Valeur moyenne de bins 10

points d'intensité 0 transformés en $2/10 = 0$; $1 \rightarrow 4/10 = 0$; $2 \rightarrow 7/10 =$

0 : les trois premiers bins sont donc regroupés dans la meme couleur

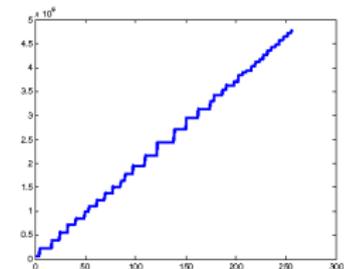
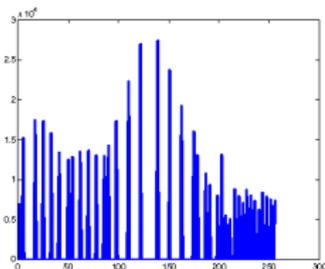
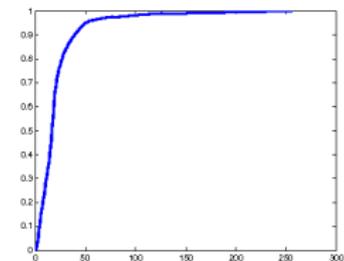
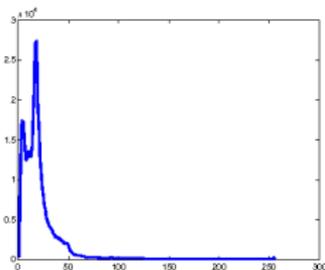
$3 \rightarrow 10/10 = 1$, $4 \rightarrow 30/10 = 3$, $5 \rightarrow 60/10 = 6$



- ▶ L'histogramme est très utilisé en image pour juger de la qualité d'une image. Si x est un niveau de gris possible, $H(x)$ est le nombre de pixels ayant le niveau de x .
- ▶ charger les images `restau1.png` et `restau2.png`. Visualiser leur histogrammes avec la fonction `imhist`. Que constatez vous ? Visualiser l'histogramme cumulé T avec la fonction `cumsum`.
- ▶ la dynamique de l'image `restau1.png` est réduite car seule la plage $[0, 100]$ de niveaux de gris est utilisé. On peut améliorer cette plage en effectuant une transformation linéaire sur les niveaux de gris. Multiplier les niveaux de gris par 2 permet ici d'utiliser la plage $[0, 200]$. Que pensez vous du résultat ?

- ▶ on souhaite améliorer la dynamique de `restau1.png`. On souhaite trouver une transformation T sur les niveaux de gris tels que IoT ait un histogramme le plus proche possible d'une loi uniforme. Cette opération est connue sous le nom d'égalisation d'histogramme. Utilisez la fonction `histeq` pour faire cette égalisation sur les deux images. Affichez les histogrammes cumulés pour ces nouvelles images.
 - ▶ A quoi est due la texture un peu granuleuse sur les images égalisées ?
 - ▶ Pourquoi n'obtient pas une loi "vraiment" uniforme ?
- ▶ on peut aussi demander avec `histeq` à avoir une égalisation de façon que l'histogramme de l'image résultat soit semblable à un histogramme donnée. Essayez ici avec vos deux images.

Quelques idées sur l'égalisation d'histogrammes



En haut : histogramme et hist cumulé pour restau1.png. En bas les histogramme égalisés

Quatrième partie IV

Quelques idées sur les applications de la
vision

- ▶ Les applications industrielles et le contrôle qualité : analyse 2D de l'image
- ▶ Compression d'images
- ▶ Les application tri-dimensionnelles de la vision : positionner et reconstruire en 3D
- ▶ Reconnaissance en vision par ordinateur (domaine en plein essor)

Contrôle qualité

Quelques exemples tirés du site d'alliance vision

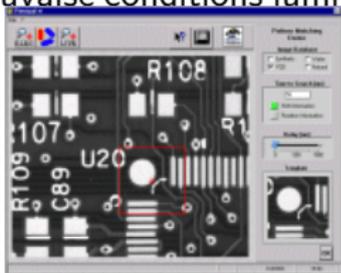
<http://www.alliancevision.fr>



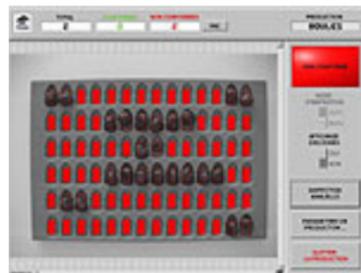
Contrôle d'aspect et des dimensions de gélules pharmaceutiques en mauvaises conditions lumineuses



Analyse de crash tests à partir de marqueurs

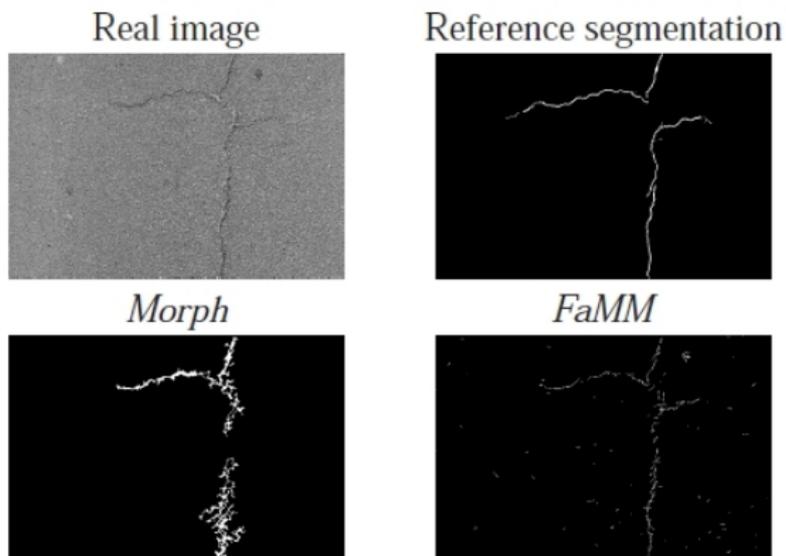


vérification de présence de composants



Contrôle de propreté

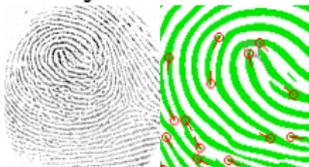
Extraction de caractéristiques en 2D



[Chambon 2010]

Exemples d'applications (2D)

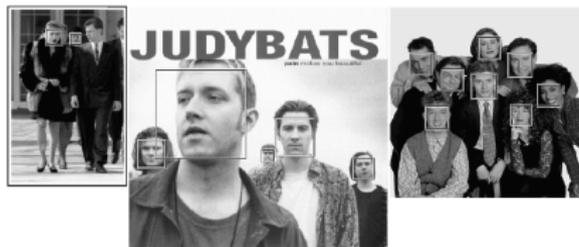
▶ Analyse 2D



▶ Construction de panoramiques [Brown & Lowe 07]



▶ Reconnaissance [Viola01]



- ▶ L'utilisation des réseaux convolutionnels profonds a bouleversé en quelques années les méthodes de reconnaissance d'objets et aussi .
- ▶ Voir le système de reconnaissance de google
<https://cloud.google.com/vision>
- ▶ Vous pouvez tester la reconnaissance avec les images dont nous nous servons en TP : chat.jpeg, venise.jpeg, restau, triGrain....

Exemples d'applications (3D)

- ▶ Reconstruction stéréoscopique [IGN]



- ▶ Structure from motion [Snavely]



- ▶ amélioration d'images/filtrage
- ▶ **détection** d'indices pertinents : contours, régions, granulométrie
- ▶ **expliquer/interpréter** le contenu des images ; classification routes, eau, prés...
- ▶ **compression** de l'information contenu dans des images **multispectrales**

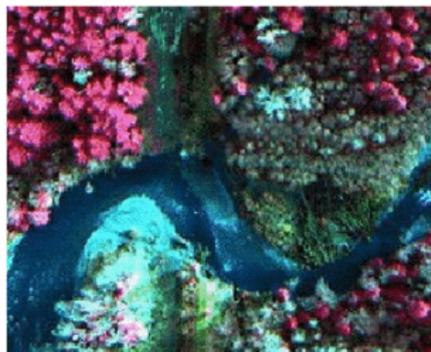


Figure 5 – Vue aérienne multispectrale représentée dans l'espace image

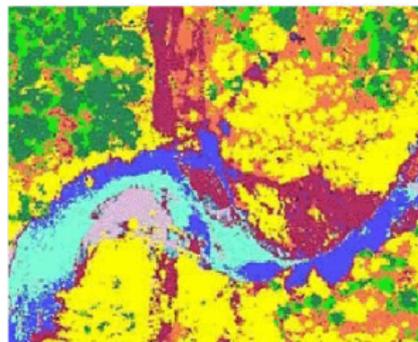
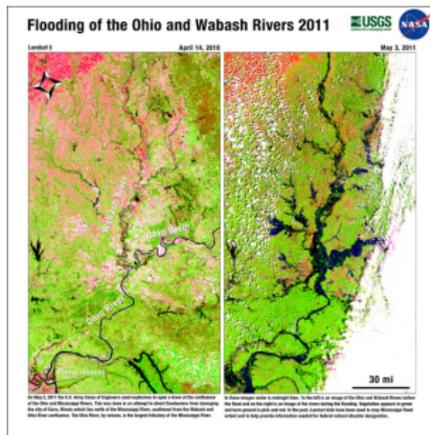


Figure 7 – Résultat de la classification.

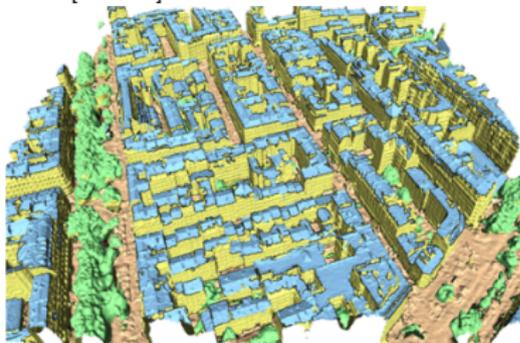
Légende : Vert foncé = conifères, Vert = taillis, Jaune = feuillus, Violet = graviers, Orange = zones sèches,

Les besoins en traitement d'images en géosciences

- ▶ analyse temporelle d'images
- ▶ reconstruction de modèle de terrains
- ▶ étiquetage sémantique de données 3D



Tiré de [Dinet05]



[Rouhani2017]

- ▶ Les bases du filtrage et de l'amélioration d'image : le couteau suisse du traitement du signal...
- ▶ Les méthodes dédiées à l'extraction d'indices et la segmentation d'images (contours, contours actifs, points d'intérêt...)
- ▶ Quelques éléments sur la reconstruction 3D
- ▶ Quelques idées sur les méthodes de type *machine learning*
 - ▶ exemples de systèmes de reconnaissance
 - ▶ comment marche un système de reconnaissance ?
- ▶ Des exemples d'applications effectives à la télédétection
 - ▶ Analyse temporelle de données
 - ▶ classification pour des images multispectrales
 - ▶ reconstruction 3D à partir d'images aériennes

Quelques ressources utiles sur le traitement d'image

- ▶ Matlab et particulièrement les toolbox : image processing, computer vision, machine learning
- ▶ opencv : bibliothèque concernant la vision par ordinateur (C,C++, android). www.opencv.org
- ▶ Image Processing on line <http://www.ipol.im/> : une revue et les codes associés
- ▶ computer vision on line <http://www.computervisiononline.com> : software, bases de données variées, livres...