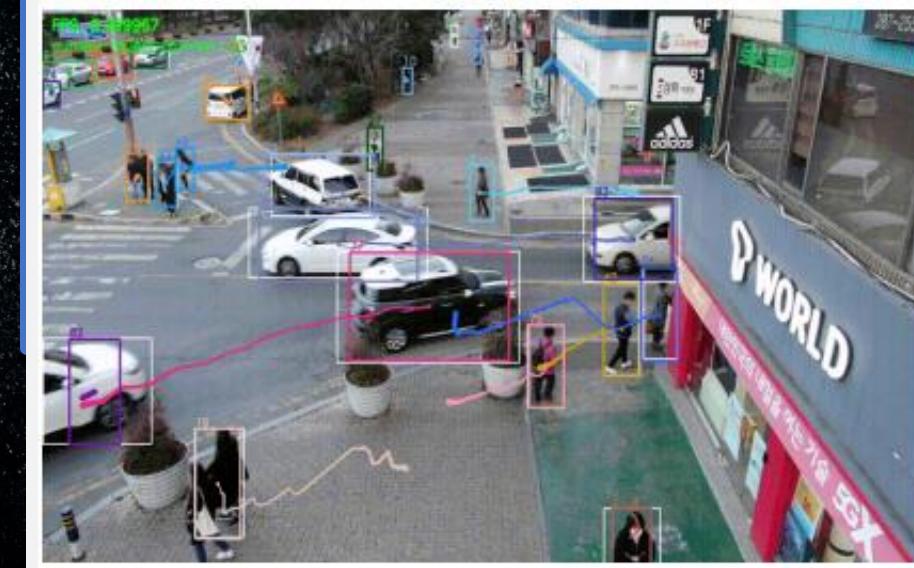
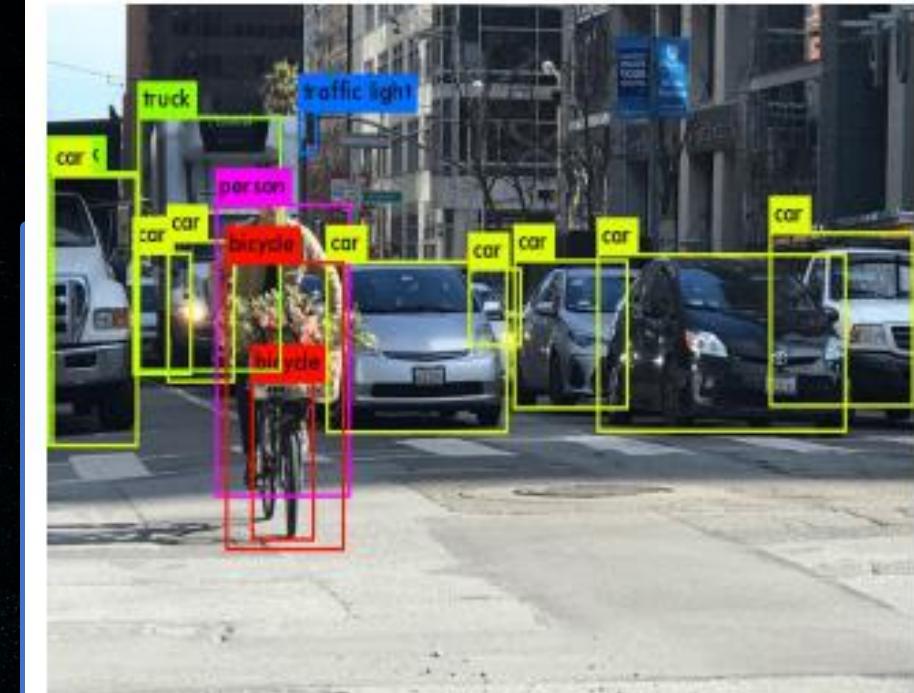


Vision par Ordinateur Computer Vision

Asma Ben Abdallah
ISIM, AU 2023-2024

Vision par ordinateur 2D
Vision par ordinateur 3D



Vision par Ordinateur

I. Introduction

- La vision par ordinateur ou computer vision :
 - permet aux ordinateurs et aux systèmes de dériver des informations significatives à partir d'images numériques, de vidéos et d'autres entrées visuelles, et de prendre des mesures ou de faire des recommandations sur la base de ces informations.
 - permet aux machines d'imiter la vision humaine
- Il s'agit d'un vaste champ pluridisciplinaire, pouvant être considéré comme une branche de l'intelligence artificielle et du Machine Learning.
- Toutefois, il est aussi possible d'utiliser des méthodes spécialisées et des algorithmes d'apprentissage général n'étant pas nécessairement liés à l'intelligence artificielle

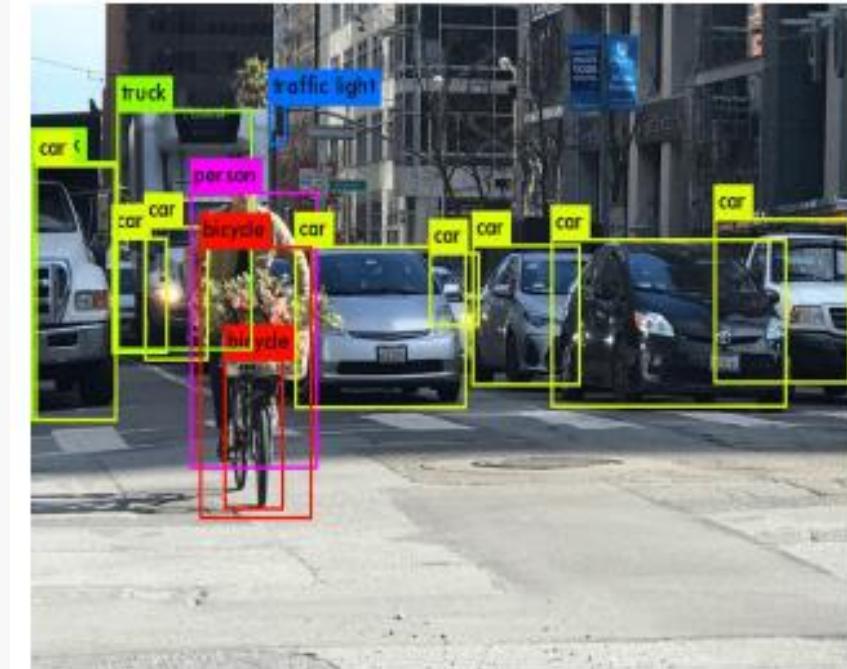


- Quel type de scène ?
- Où se trouve la voiture?
- A quelle distance se trouve la maison?

Vision par ordinateur

I. Introduction

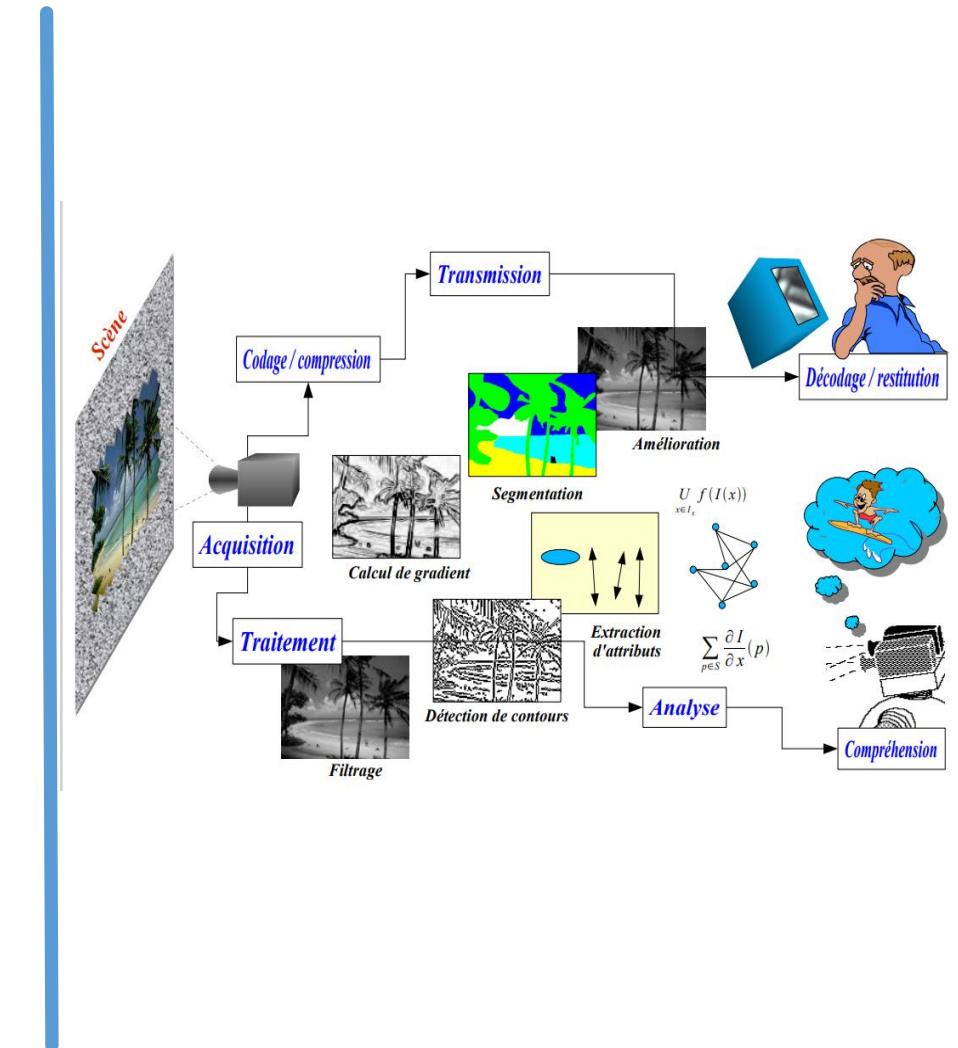
- La vision par ordinateur se présente comme un outil basé sur l'IA capable de traiter, d'analyser, de comprendre, une image et de traiter les informations qui en découlent.
- C'est l'équivalent, en termes d'IA, des yeux humains et de la capacité de notre cerveau.
- Elle associe caméras, edge computing, informatique en nuage, logiciels et intelligence artificielle (IA) pour aider les systèmes à «voir» et à identifier les objets.
- Concrètement, l'idée est de transmettre à une machine des informations sur le monde réel à partir des données d'une image observée.



Vision par ordinateur

I. Introduction

- Les tâches de vision par ordinateur comprennent des procédés pour acquérir, traiter, analyser et « comprendre » des images numériques, et extraire des données afin de produire des informations numériques ou symboliques, par ex. sous forme de décisions.
- Les données d'image peuvent prendre de nombreuses formes, telles que des séquences vidéo, des vues de plusieurs caméras, des données multidimensionnelles à partir d'un scanner 3D ou d'un appareil de numérisation médical.



Vision par Ordinateur

II. Vision par ordinateur : Domaines d'application

Reconnaissance de caractères (OCR):

- Traitement de texte
- Tri Postal
- Reconnaissance d'un matricule



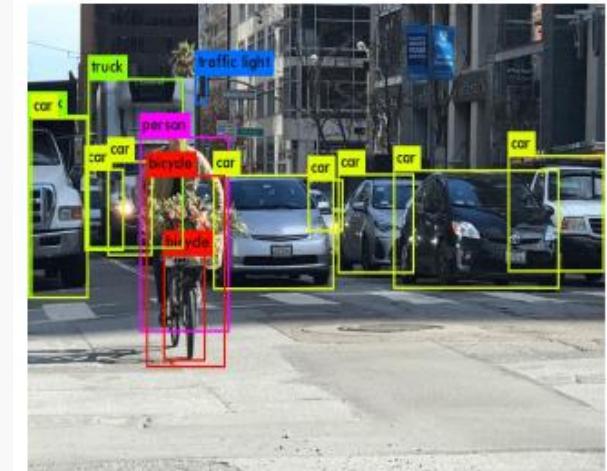
Reconnaissance faciale



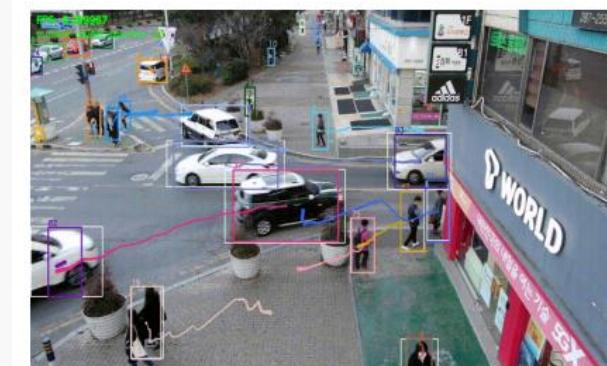
Détection d'objet

Suivi d'objet

► Détection d'objet



Suivi d'objet



Vision par Ordinateur

II. Vision par ordinateur : Domaines d'application

□ Identification

Reconnaissance d'empreinte,
Voix, Visage,
Œil, ...



□ Recherche d'une image



□ Effets Spéciaux

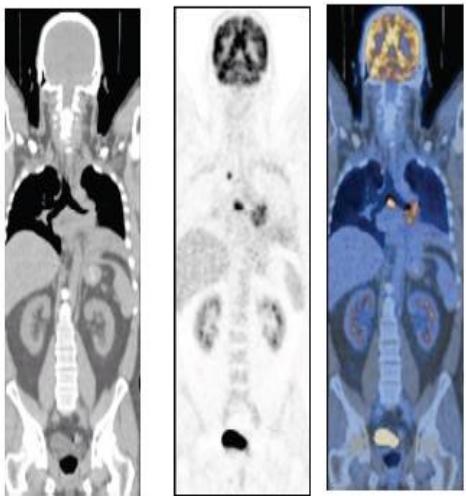
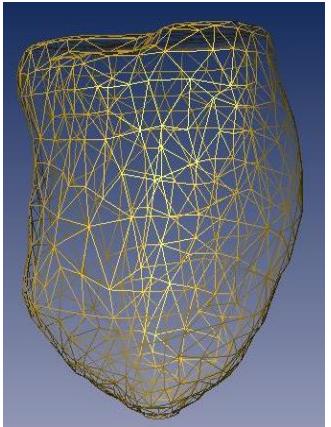
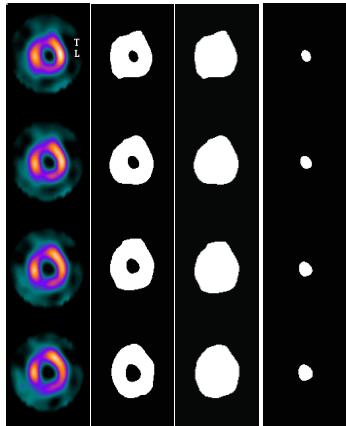
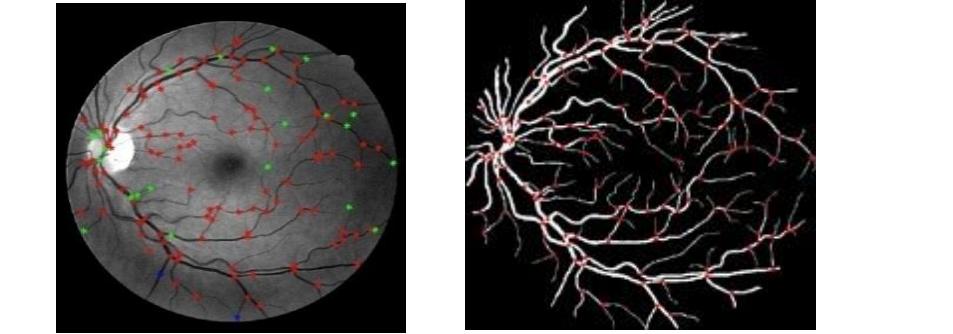


□ Jeux vidéo



Vision par Ordinateur

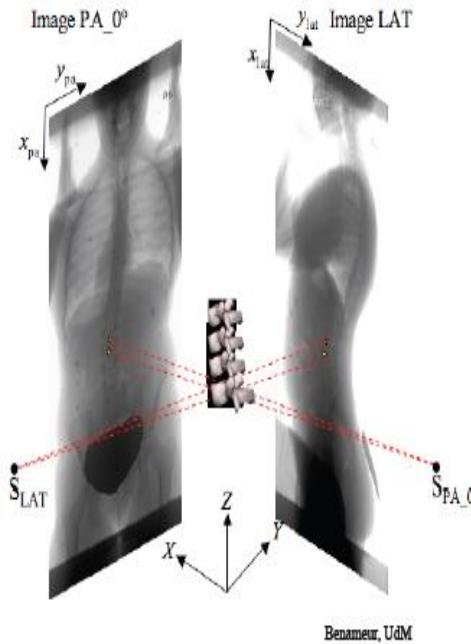
Imagerie médicale : Aide au diagnostic, reconstruction 3D, ...



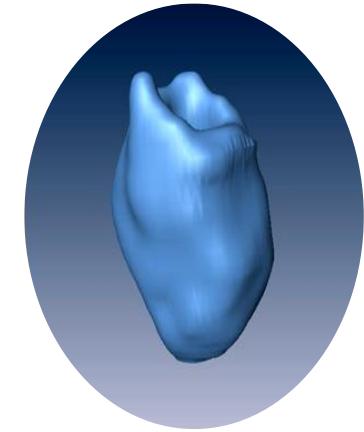
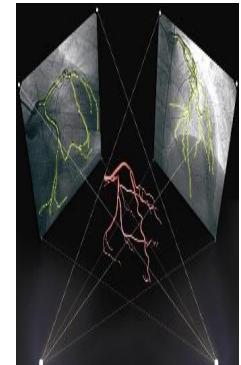
IRM
Anatomie

Petscan
Activité
métabolique

Exemple
de fusion



Reconstruction 3D



Vision par Ordinateur

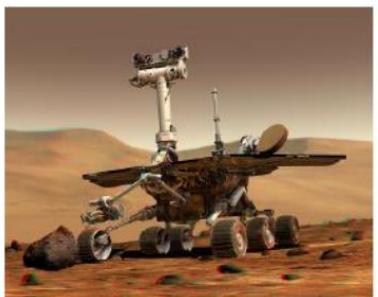
II. Vision par ordinateur : Domaines d'application

□ Robotique industrielle

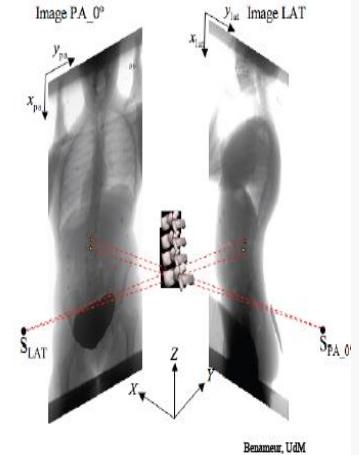
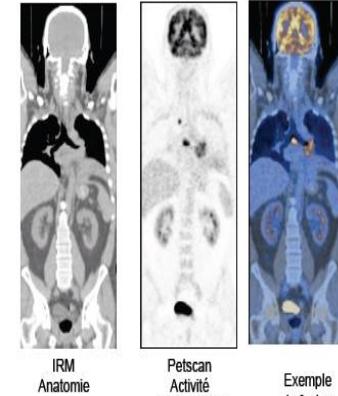
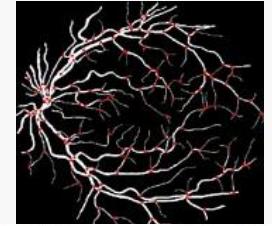
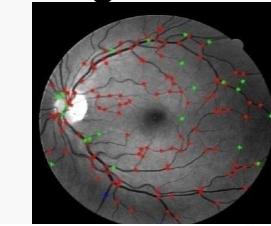


□ Robots mobiles

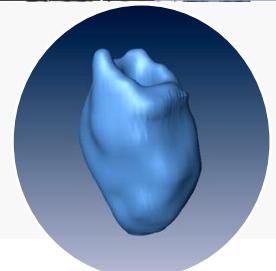
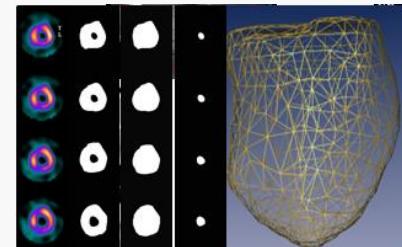
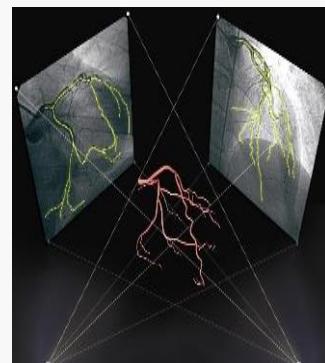
- évitement d'obstacles
- découverte d'un environnement inconnu et construction d'une carte (exploration)
- conduite automatique / assistée



□ Imagerie Médicale



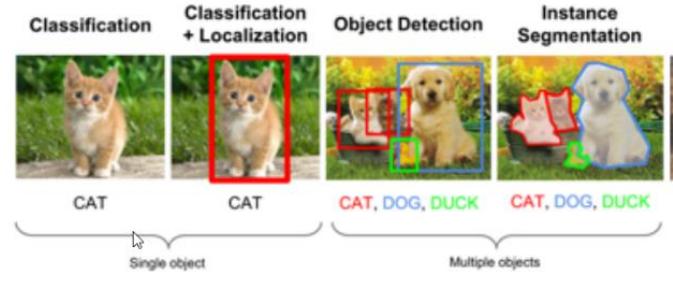
Reconstruction 3D



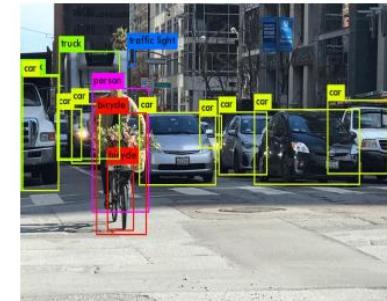
Vision par Ordinateur 2D

III. Exemple de tâches de vision 2D-3D

1. **Classification d'images** : Attribuer une catégorie ou un label prédéfini à une image. Par exemple, identifier si une image contient un chien, un chat, une voiture, etc.
2. **Détection d'objets** : Identifier et localiser les objets spécifiques dans une image. Cela implique de dessiner des boîtes englobantes autour des objets et d'identifier leur classe.
3. **Suivi d'objets** : Suivre le mouvement des objets dans une séquence d'images, souvent utilisé dans la surveillance vidéo.
4. **Segmentation sémantique** : Attribuer un label à chaque pixel de l'image, en identifiant les régions correspondant à différents objets ou structures.

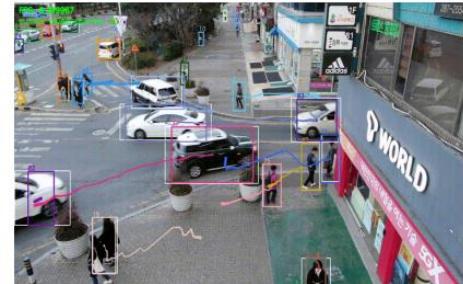


► Détection d'objet



Requires Classification

Suivi d'objet



Vision par Ordinateur 2D

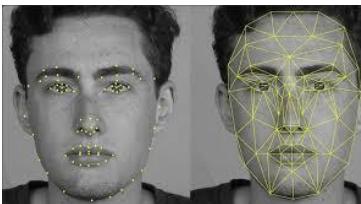
III. Exemple de tâches de vision 2D-3D

5. Restauration d'image : exploite le Machine Learning pour supprimer le " bruit " (grain, flou...) sur des photos.

6. Reconstruction de scène : permet de créer un modèle 3D d'une scène à partir d'images et de vidéos.

7. Estimation de pose d'un objet ou d'une personne dans l'espace en fonction de l'image afin d'en extraire un modèle biomécanique. Par exemple, la détection des articulations du corps humain.

8. Reconnaissance faciale : Identifier et reconnaître les visages humains dans une image. Par exemple, utilisation pour l'authentification biométrique.

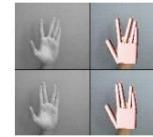


MULTIMEDIA

Reconnaissance d'action dans un match de tennis pour l'indexation vidéo (INRIA)

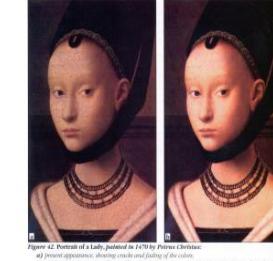


Reconnaissance et synthèse de geste pour le codage et la transmission du langage signé (INT Évry)



codage et compression

recherche d'images dans une base de données à partir d'une requête visuelle (ENSEA Cergy-Pontoise)



restauration d'images

indexation d'images



MÉDECINE

modèle 3D de cerveau reconstitué à partir de coupes d'images acquises par résonance magnétique (Univ. Québec)



téléchirurgie et ChAO

estimation de mouvement sur séquence d'images échographiques (INRIA)



Aide au diagnostic

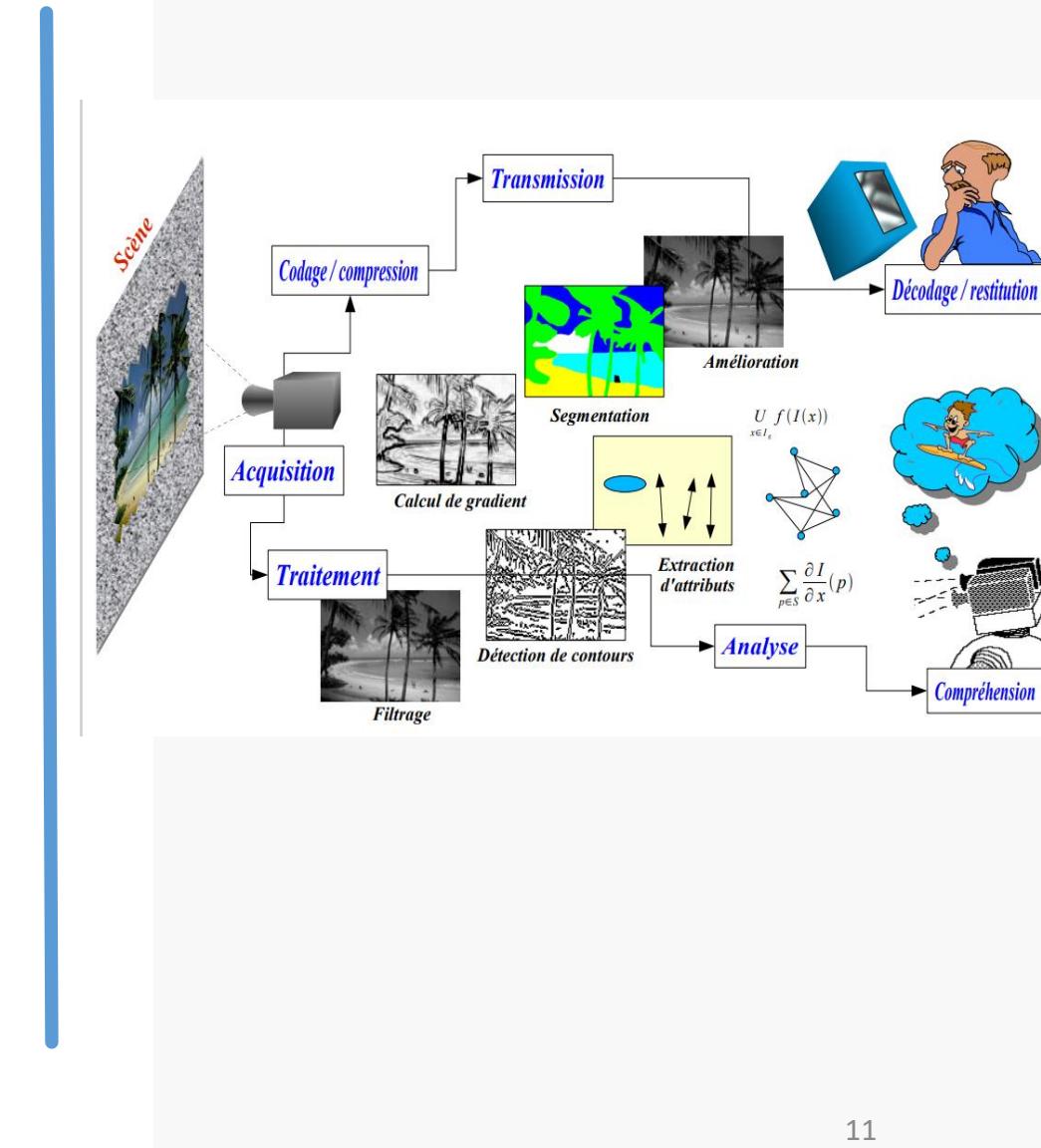


Vision par Ordinateur

IV. Système de Vision par Ordinateur

Connaissances et technologies qui permettent de concevoir des machines qui peuvent « voir »

- **Premier niveau : vision**
acquiert une image grâce un capteur
- **Deuxième niveau (bas niveau) : traitement de l'image**
modifie le contenu de l'image afin de mettre en évidence des éléments d'intérêt (objets, contours)
- **Troisième niveau (Haut Niveau) : reconnaissance**
utilise des techniques d'intelligence artificielle pour identifier des formes connues dans l'image



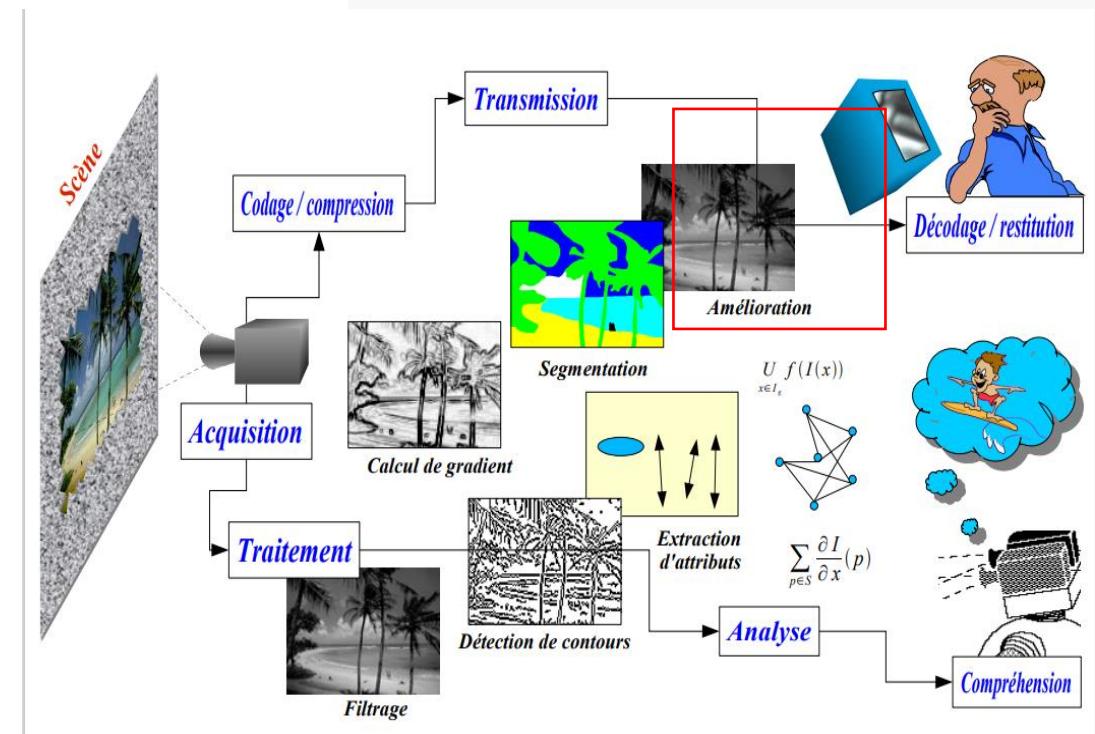
Partie 1 : Vision 2D

Rappel

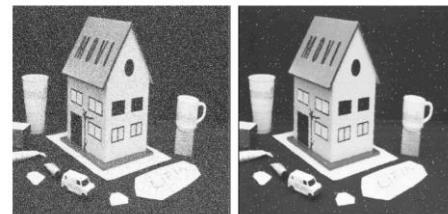
- Amélioration d'images :
 - histogrammes,
 - lissage, (filtrage médian, moyenneur, Gaussien, ...)

- Contours d'une image de dimension quelconque :
 - extrema locaux du gradient
 - ou zéros du laplacien

- Segmentation d'une image



Filtrage médian : 3x3



Gradient par Sobel



Vision 2D

Segmentation

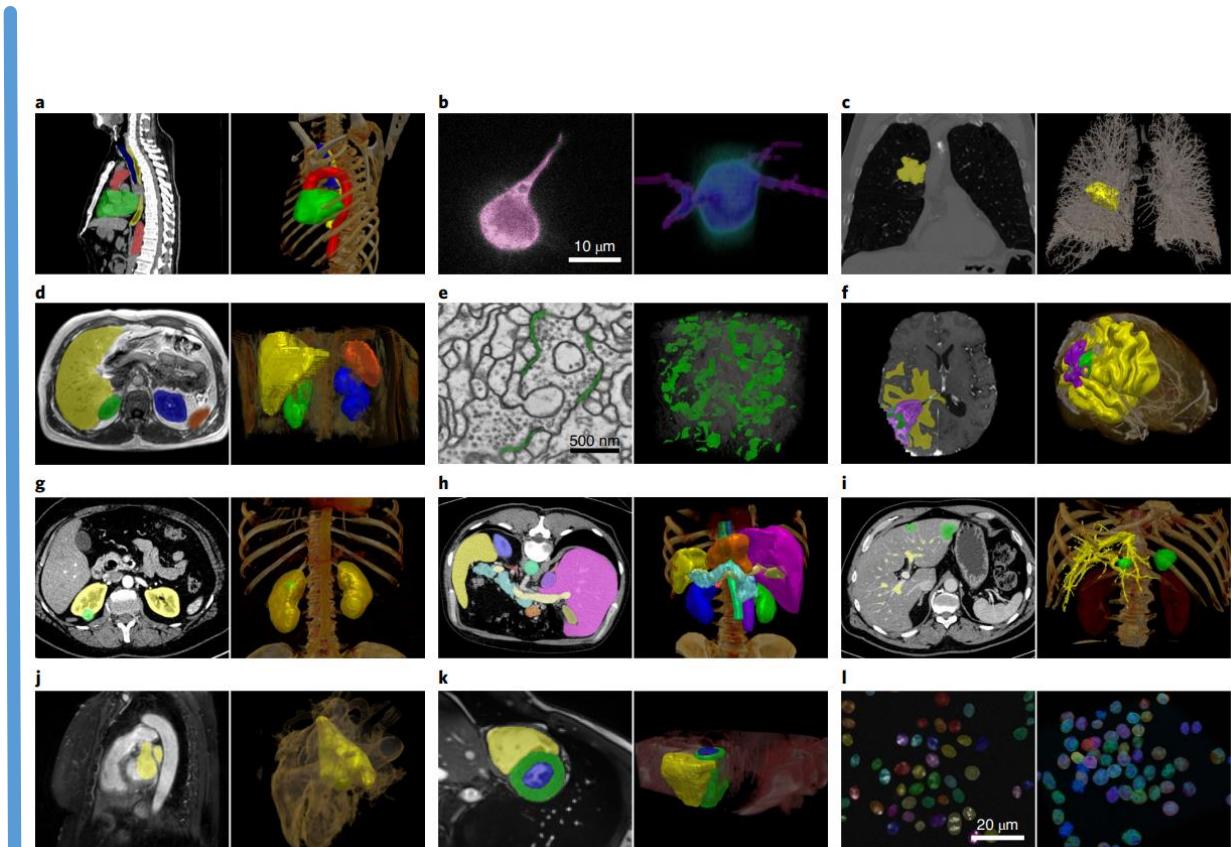
Plan

- Qu'est ce que la segmentation
- Approches de la segmentation
- Méthodes de segmentation
 - Seuillage
 - Croissance de régions (Region growing)
 - Décomposition et fusion (Split & Merge)
 - Segmentation basée Deep Learning

I. Qu'est ce que la segmentation

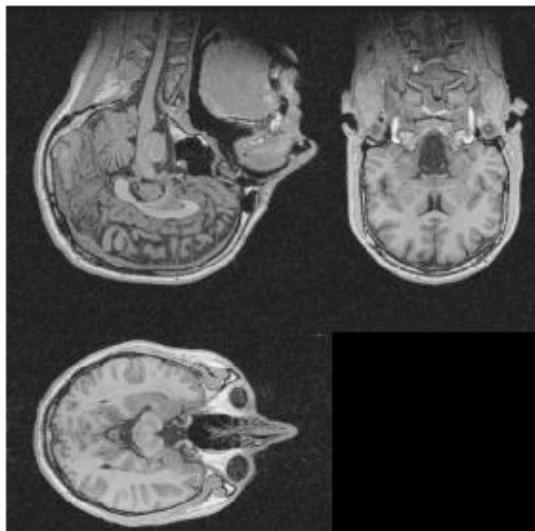
- La segmentation d'images est l'un des problèmes phares du traitement d'images. Elle consiste à partitionner l'image en un ensemble de régions connexes R_i .

- 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.

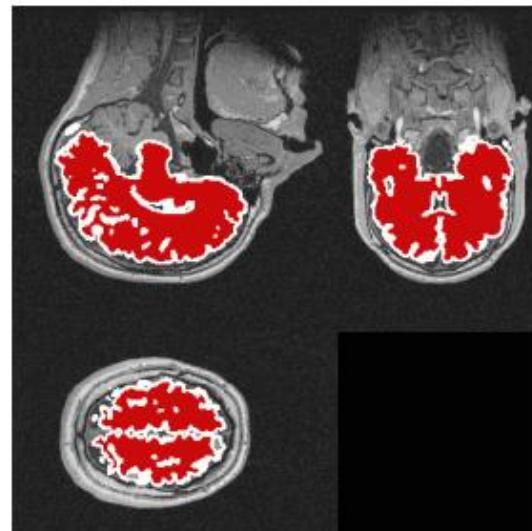


Segmentation: Exemple d'applications

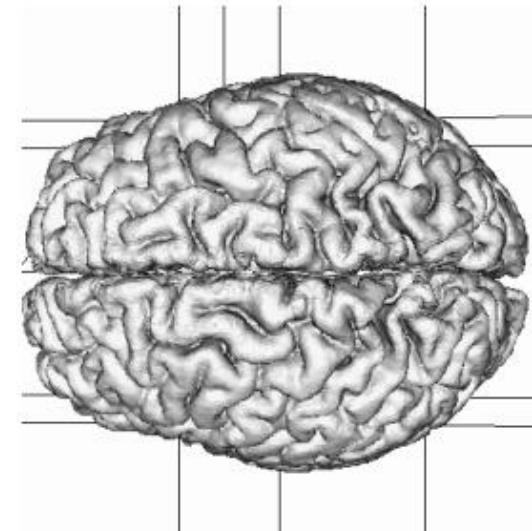
Reconstruction 3D du cortex cérébral
(Approche contour : modèle déformable)



Images 3D (3 coupes)



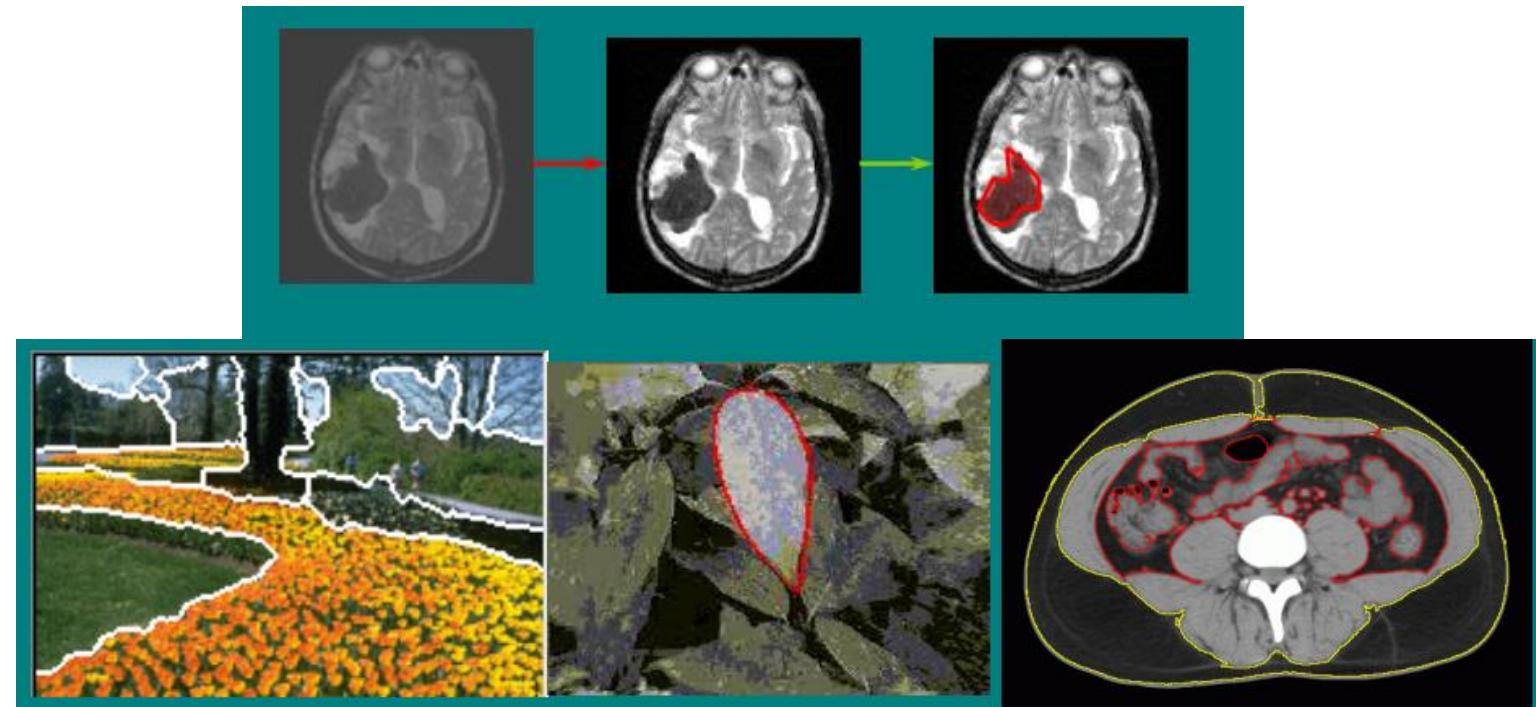
Segmentation



Reconstruction 3D

I. Qu'est ce que la segmentation

- La segmentation d'un point de vue **formel**, est un traitement de bas niveau qui consiste à **créer une partition** d'une image A en sous ensemble de régions R_i .
- La région étant un ensemble **connexe** ayant des caractéristiques communes (intensité, texture, ...)
- La segmentation est liée à la reconnaissance (quels objets voit-on dans l'image?)

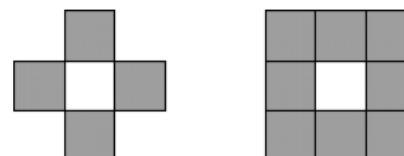


I. Qu'est ce que la segmentation-Notion de connexité

- La connexité est une propriété de liaison entre deux pixels considérés de la même région dans une image.
- **La 4-connexité** : deux pixels sont 4-connexes s'ils se suivent sur la même ligne ou la même colonne.
 - Exemple : Dans l'image à droite, un exemple de paires de pixels connexes : (A,B), (A,C), (A,D) et (A,E).
 - **A et F ne sont pas 4-connexes.**
- **La 8-connexité** : deux pixels sont 8-connexes s'ils se suivent verticalement, horizontalement ou en diagonale.
 - Dans l'exemple ci-dessous, A et F sont 8-connexes.

O	B	O
C	A	D
O	E	F

connexités 2D : 4-connexité / 8-connexité



I. Qu'est ce que la segmentation

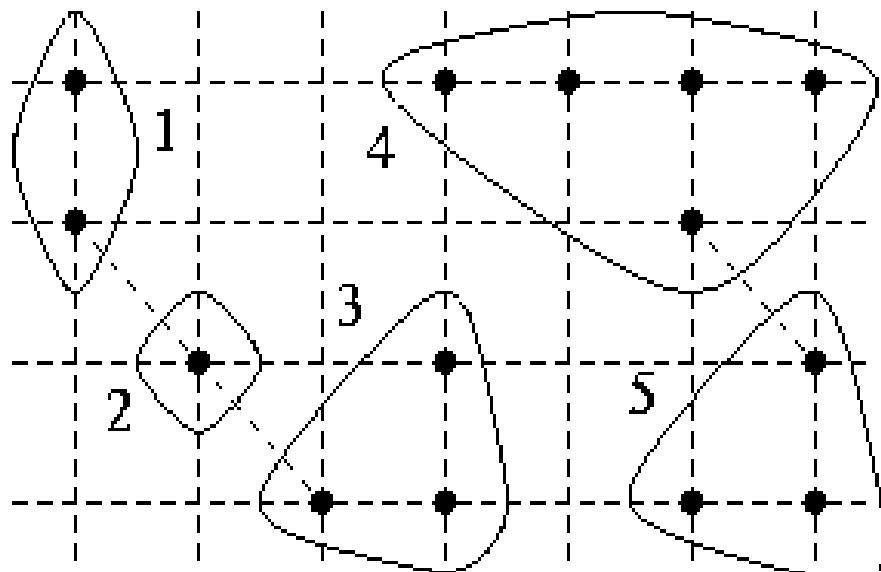
NOTION DE CONNEXITÉ :

- On appelle une *composante k-connexe* de l'image, un ensemble de pixels tels que, si l'on prend 2 pixels quelconques dans cet ensemble, il est possible de trouver un chemin k-connexe qui les relie (e,g k=4, 8).
- Exemple (A,B,C,D,E,F) est une composante connexe

O	B	O
C	A	D
O	E	F

I. Qu'est ce que la segmentation

COMPOSANTE CONNEXE : EXEMPLE-2



- les composantes 4-connexes sont les ensembles numérotés de 1 à 5,
- les composantes 8-connexes sont :
 - d'une part la réunion des ensembles 1 à 3,
 - d'autre part la réunion des ensembles 4 et 5.

étant donné une famille d'ensembles k -connexes ayant un pixel p en commun, la réunion de ces ensembles est k -connexe.

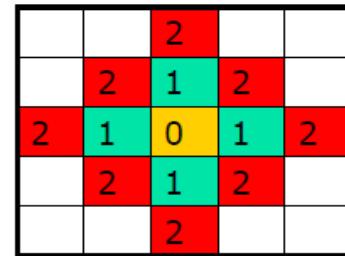
I. Qu'est ce que la segmentation

NOTION DE CONNEXITÉ, DISTANCE

- **Distance D_4 (distance de Manhattan)**

$$D_4(p,q) = |x-s| + |y-t|$$

- Forme un diamant centré sur (x,y)
- Ex: $D_4 \leq 2$



- **Distance D_8 (distance de l'échiquier)**

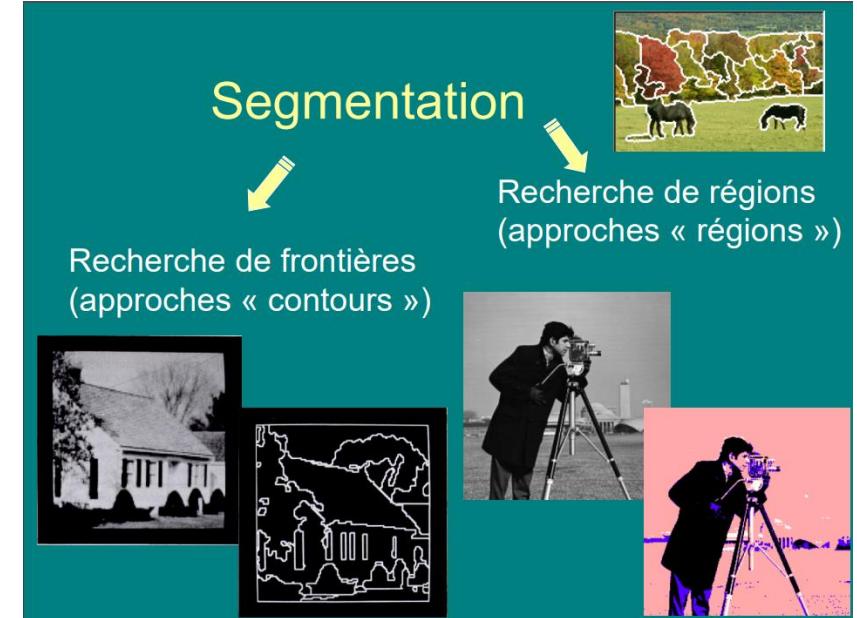
$$D_8(p,q) = \text{maximum } (|x-s|, |y-t|)$$

- Forme un carré centré sur (x,y)
- Ex: $D_8 \leq 2$



II. Segmentation Région/Contours

- La segmentation fait référence aux notions de *différence* et de *similarité* comme les perçoit le système visuel humain et ceci donne naissance à deux approches principalement :
 - approche contour
 - et approche région.
- Ces deux approches sont duales du fait que chaque région possède un contour et qu'un contour délimite nécessairement une région.



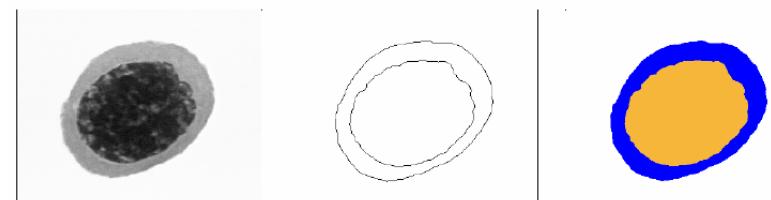
II. Segmentation Région/Contours

La segmentation est basée sur:

- Les discontinuités : **contours**
 - Rechercher les discontinuités entre régions (les changements abruptes, frontières entre régions...)
- Les zones homogènes : **régions**
 - Rechercher un critère d'homogénéité entre régions (Mêmes couleurs, textures, intensités, ...)
- La segmentation est le découpage d'une image en différentes **régions et/ou contours**. Un contour fermé est équivalent à une région

Un contour fermé est équivalent à une région.

Dualité Régions/Contours



II. Segmentation Région/Contours

Étiqueter \equiv partitionner

Définition de [Horowitz75]

X : domaine de l'image /

P : prédicat défini sur l'ensemble des parties de X , dépend de /

segmentation de X : $(S_i)_{i=1..n}$, sous-ensembles de X tels que

$X = \cup_{i=1}^n S_i$ (Partition de X)

$\forall i \in 1..n$, S_i est connexe et $P(S_i) = vrai$

$\forall i, j \in 1..n$, S_i adjacent à S_j et $i \neq j \Rightarrow P(S_i \cup S_j) = faux$

$\forall (i \neq j) \quad R_i \cap R_j = \emptyset$

$\forall (i) \quad R_i \neq \emptyset$

Exemples de prédicat d'homogénéité :

- $P(R) = Vrai \Leftrightarrow \sigma_R < 5$
- $P(R) = Vrai \Leftrightarrow \forall p \in R, |I(p) - \mu_R| < 10$

Activer Window

III. Méthodes de segmentation

Plusieurs méthodes de segmentation:

- Seuillage : segmentation par histogramme
- Croissance de région
- Décomposition et fusion (Split & merge)
- Segmentation basée DL

III.1 Segmentation par seuillage

- Le seuillage est une méthode simple et très populaire pour le traitement des images numériques.
- Ce n'est pas une méthode de segmentation en régions
 - **Approche pixel (ni région ni contour)**
 - Utilisée souvent en segmentation (avec post-traitements)
- Le seuillage peut être :
 - Global : un seuil pour toute l'image
 - Local : un seuil pour une portion de l'image

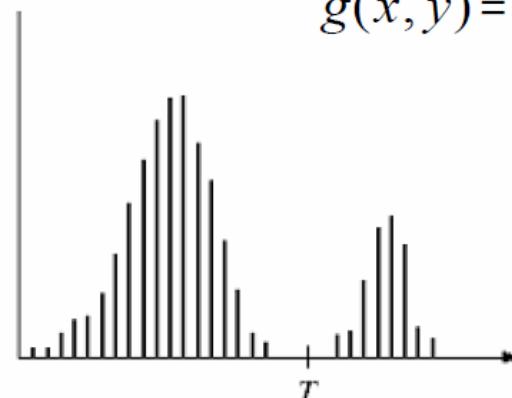
III.1 Segmentation par seuillage

Principe : trouver les différents modes de l'histogramme, représentatifs d'autant de classes d'objets dans l'image.

- On peut rechercher par exemple le(s) seuil(s) à partir d'une analyse de l'histogramme.
- Exemple : Seuillage d'histogramme simple ou multi-seuils
- Le résultat du seuillage est une image binaire 0 ou 1 (qu'on transforme parfois en 0:255 pour l'affichage)

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases}$$

Problème : choix du seuil !

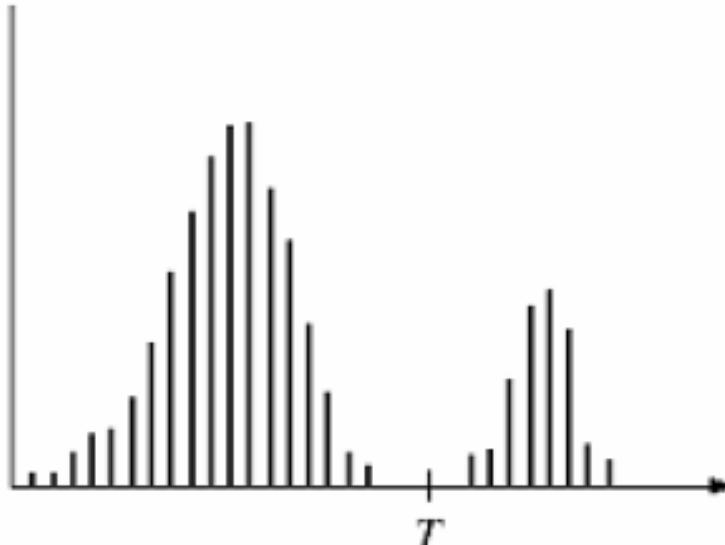


Segmentation par seuillage

Comment définir le seuil?

- Comment trouver le bon seuil (T) ?
 - Une valeur obtenue par tests
 - La valeur moyenne des tons de gris
 - La valeur médiane entre le ton maximum et le ton minimum
 - Une valeur qui balance les deux sections de l'histogramme

Il existe des algorithmes automatiques pour trouver le seuil !



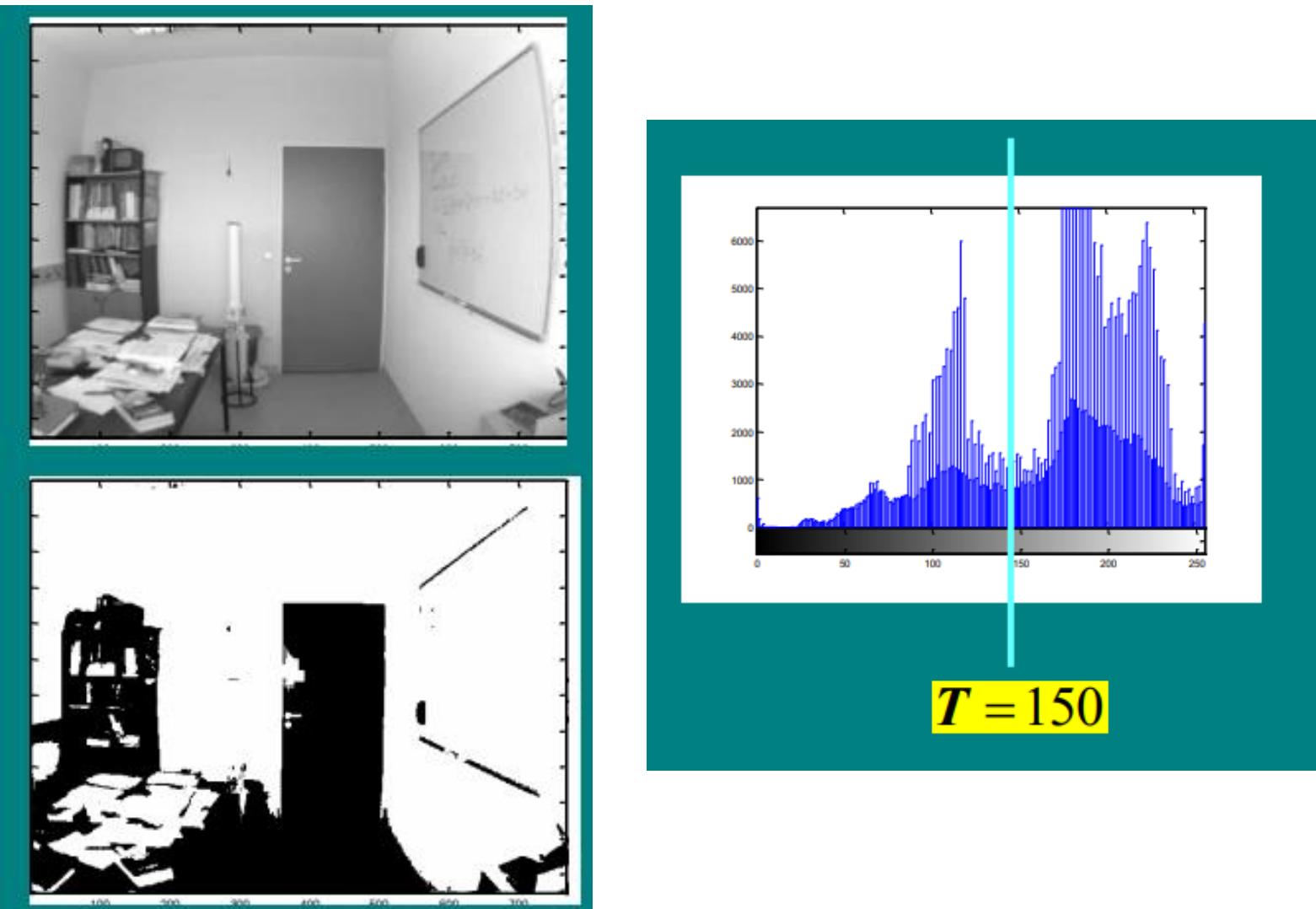
Segmentation par seuillage

Seuillage global automatique

- Il existe plusieurs méthodes globales automatiques qui permettent de trouver un seuil à partir d'un histogramme
 - *Otsu, Kittler, ...*
- Mais chaque méthode donne un seuil différent
 - *Quel est le bon seuil ?*
- Il faut tester pour chaque nouvelle application

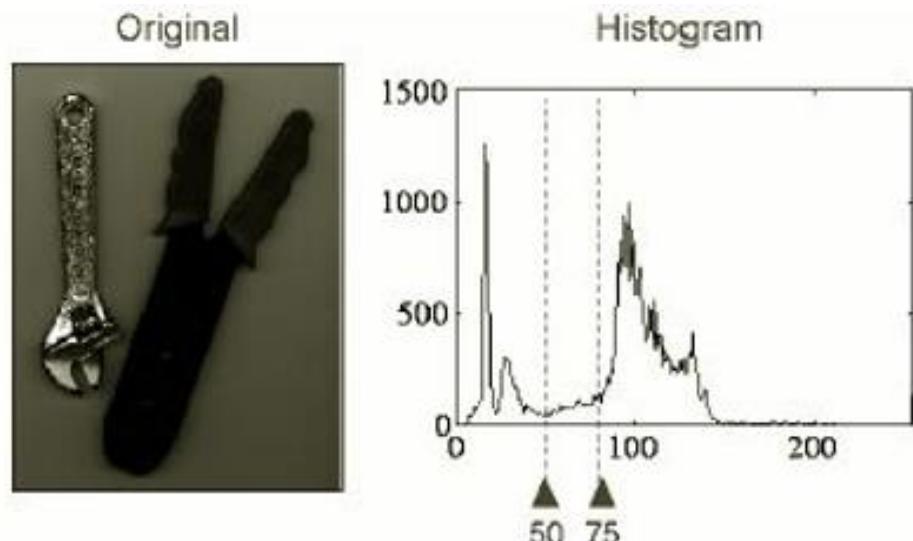
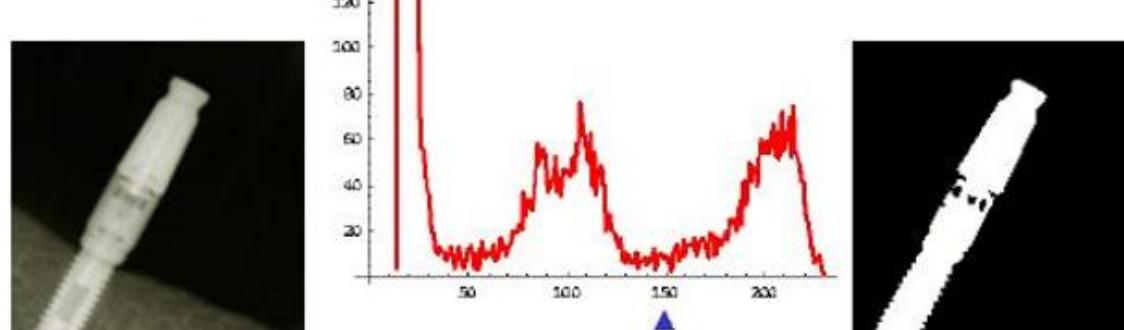
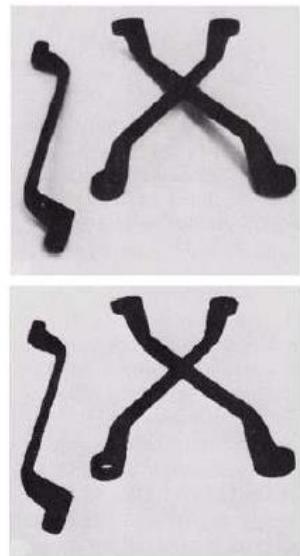
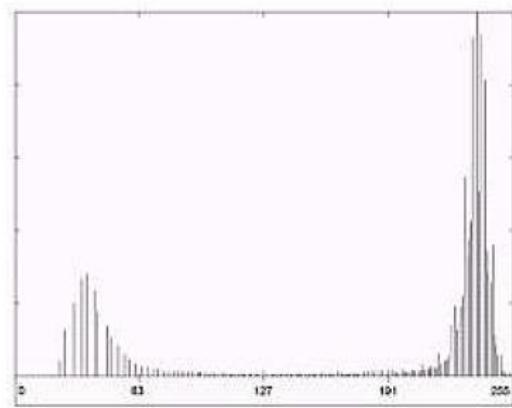
Segmentation par seuillage

Exemple : Seuillage simple



Segmentation par seuillage

Exemples : Seuillage simple



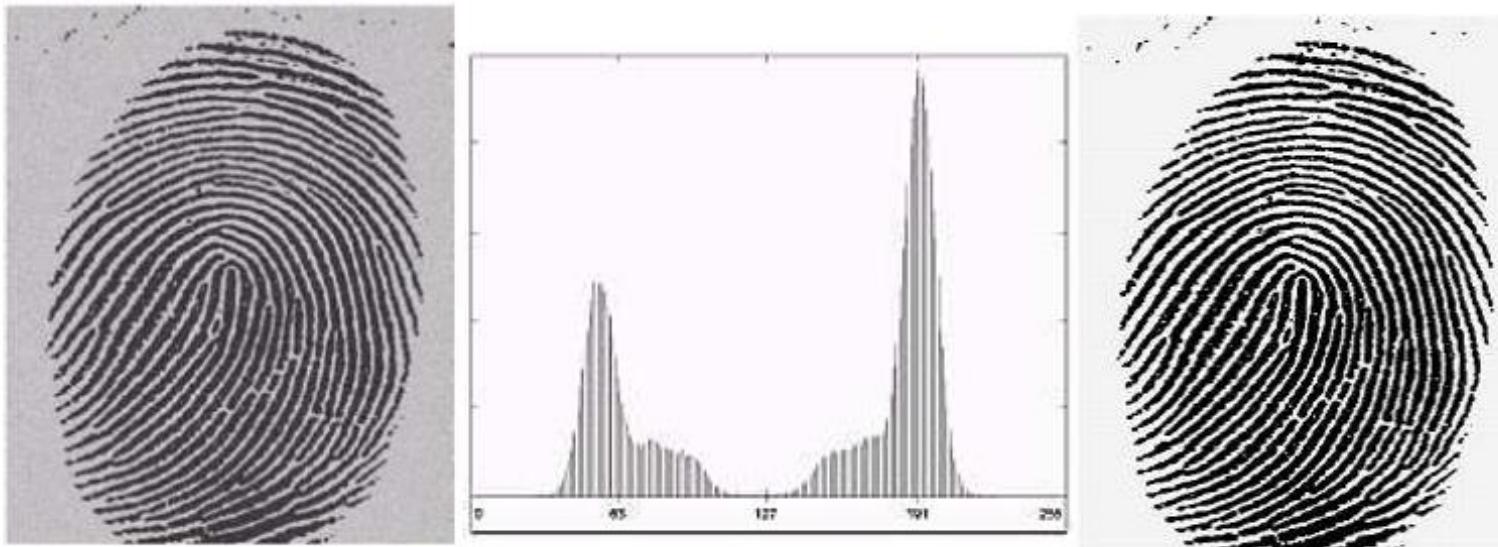
Threshold = 50



Threshold = 75

Seuillage global automatique : exemple : Seuillage simple

Seuil trouvé par l'algorithme : $T = 125$

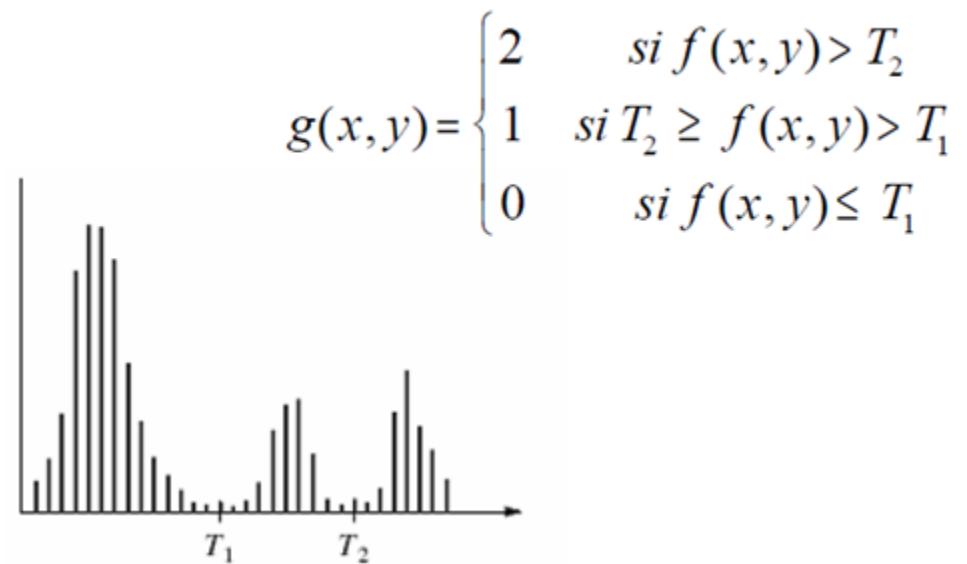


III.1 SEGMENTATION PAR SEUILLAGE

Segmentation par seuillage multi-seuils

- n seuils pour séparer l'image en $n+1$ classes :
 - Si $valeur(pixel) < seuil_1$
 - alors $valeur(pixel) = 0$
 - Si $valeur(pixel) \geq seuil_1 \ \&\& \ valeur(pixel) < seuil_2$
 - alors $valeur(pixel) = 1$
 - ...
 - Si $valeur(pixel) \geq seuil_n$
 - alors $valeur(pixel) = n$
- **Problèmes** : Combien de seuils + Valeurs des seuils !

Seuillage d'histogramme multiple
Cas de deux seuils



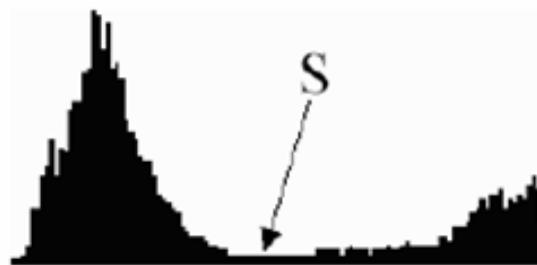
Segmentation en pixels

Segmentation par seuillage :

- + Universel, temps réel, simplicité
- + Fonctionne bien sur des histogrammes multi-modaux

- Il faut connaître le nombre de classes
- Apparition de faux éléments (aucune prise en compte de la composante spatiale)
- Le seuillage est une opération sur les pixels : ne produit **pas forcément des régions connexes**,

- Il existe des méthodes de segmentation en régions



2 seuils pour 3 classes

III.2 Segmentation en régions

Objectif : Trouver la partition de l'image tenant compte du double critère : Homogénéité de la région R_i et Voisinage

Soit X le domaine de l'image et f la fonction qui associe à chaque pixel une valeur $f(x, y)$. Si nous définissons un prédictat P sur l'ensemble des parties de X , la segmentation de X est définie comme une partition de X en n sous-ensembles $\{R_1, \dots, R_n\}$ tels que :

1. $X = \bigsqcup_{i=1}^m R_i$
 2. $\forall i \in \{1, \dots, n\} R_i$ est connexe.
 3. $\forall i \in \{1, \dots, n\} P(R_i) = \text{vrai}$
 4. $\forall i, j \in \{1, \dots, n\}^2 / R_i$ est adjacent à R_j et $i \neq j \Rightarrow P(R_i \cup R_j) = \text{faux}$
- où \bigsqcup représente une union d'ensemble disjoints.

1 : tout pixel de l'image appartient à une région et une seule.

$$\begin{aligned}\forall (i \neq j) R_i \cap R_j &= \emptyset \\ \forall (i) R_i &\neq \emptyset\end{aligned}$$

2 : toute région doit être connexe. La connexité des régions est induite par le voisinage défini sur l'image.

3 : chaque région doit être homogène.

4 : la fusion de deux régions ne doit pas être homogène

Exemples de prédictat d'homogénéité :

- $P(R) = \text{Vrai} \Leftrightarrow \sigma_R < 5$
- $P(R) = \text{Vrai} \Leftrightarrow \forall p \in R, |I(p) - \mu_R| < 10$

III.2 Segmentation en régions

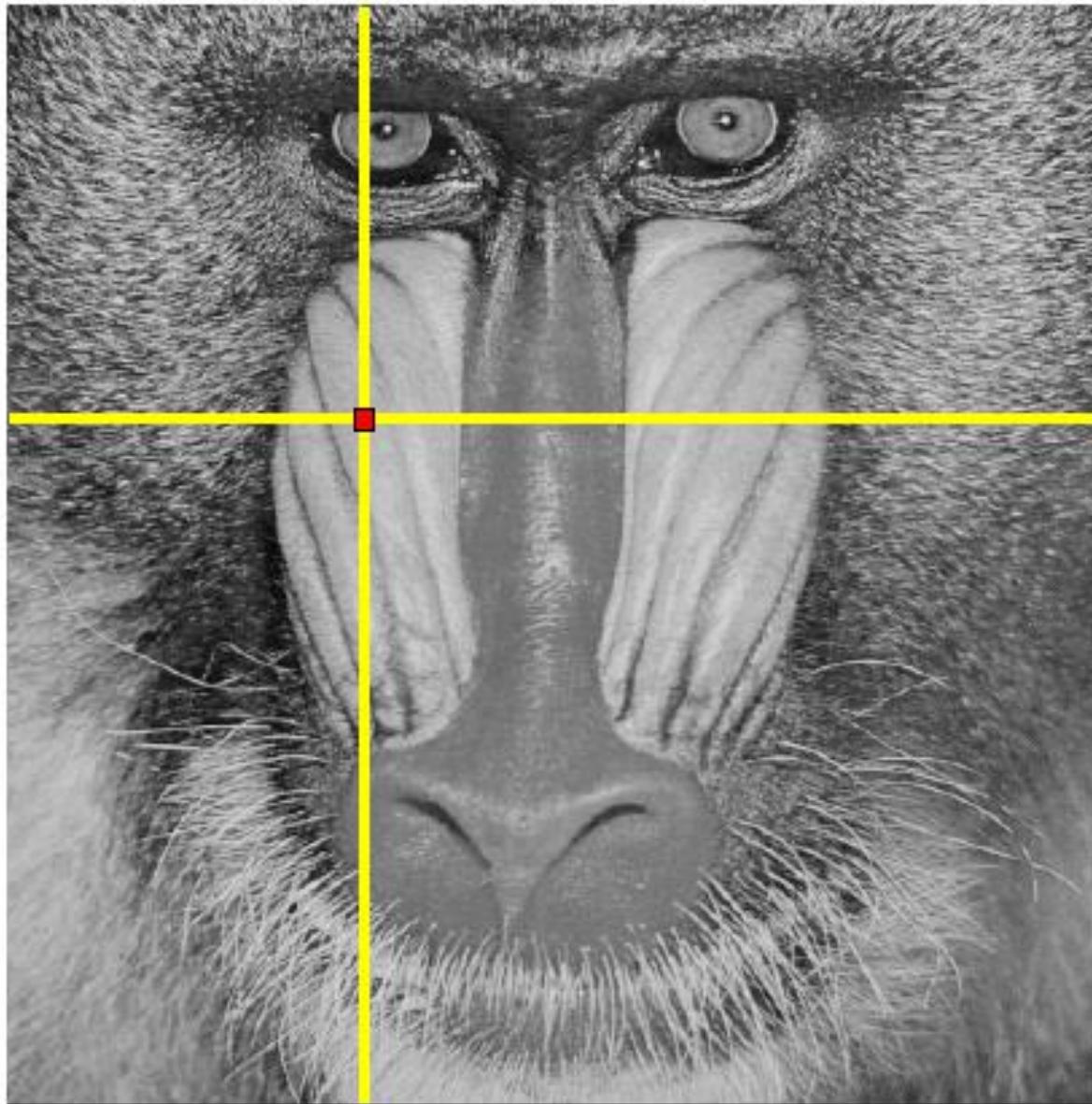
Méthode croissance des régions (Region growing)

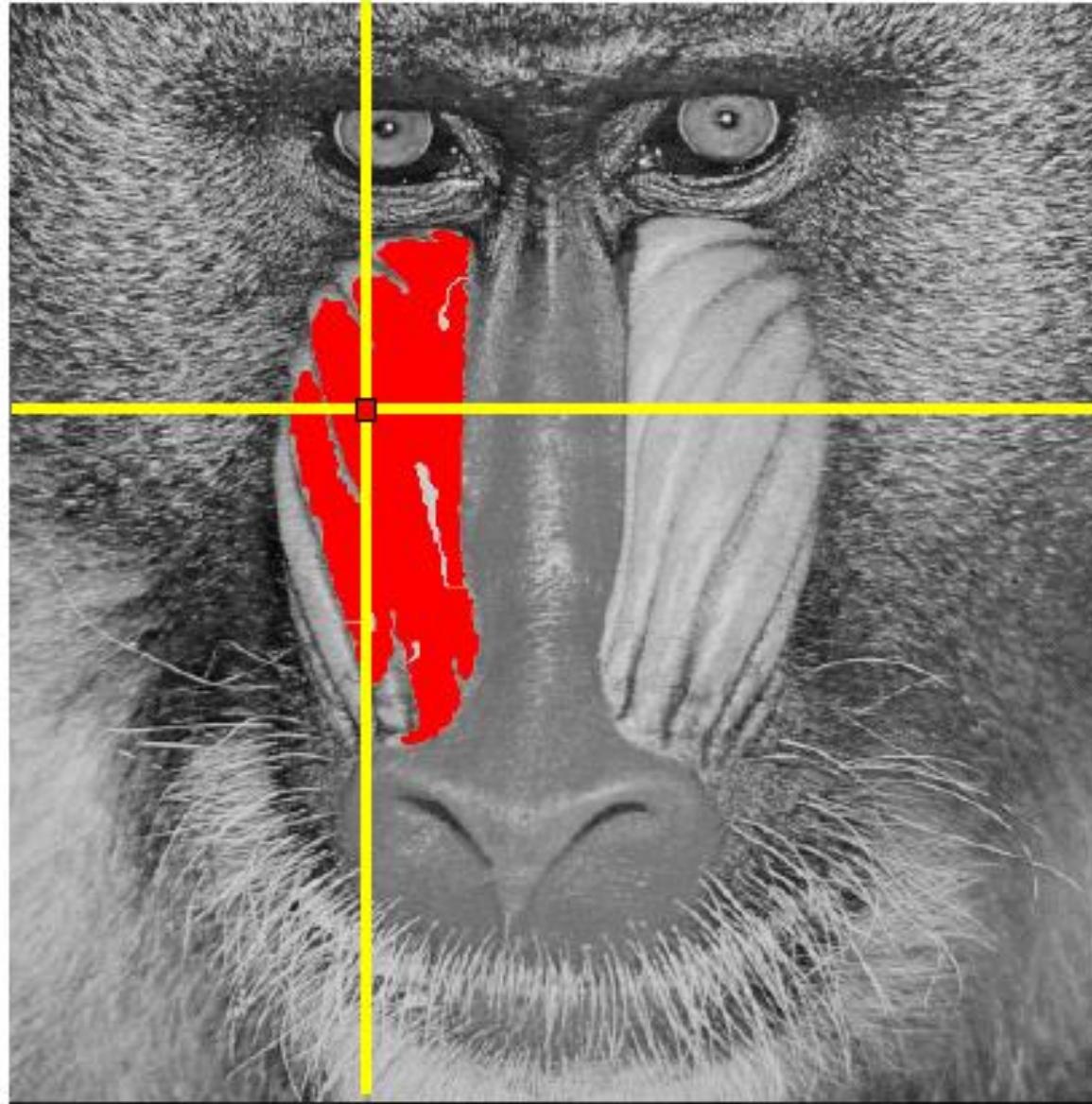
- Dans cette technique, on construit de **façon itérative** une région partir d'un ensemble initial (qui peut être réduit à un pixel), qu'on fait **grossir** en ajoutant des pixels en fonction de certaines conditions.
- Ces conditions peuvent être :
 - relatives au niveau de gris (ou à la couleur) du nouveau pixel candidat, comparé au niveau de gris (ou couleur) moyen de la région, ...
 - Adjacence de pixels

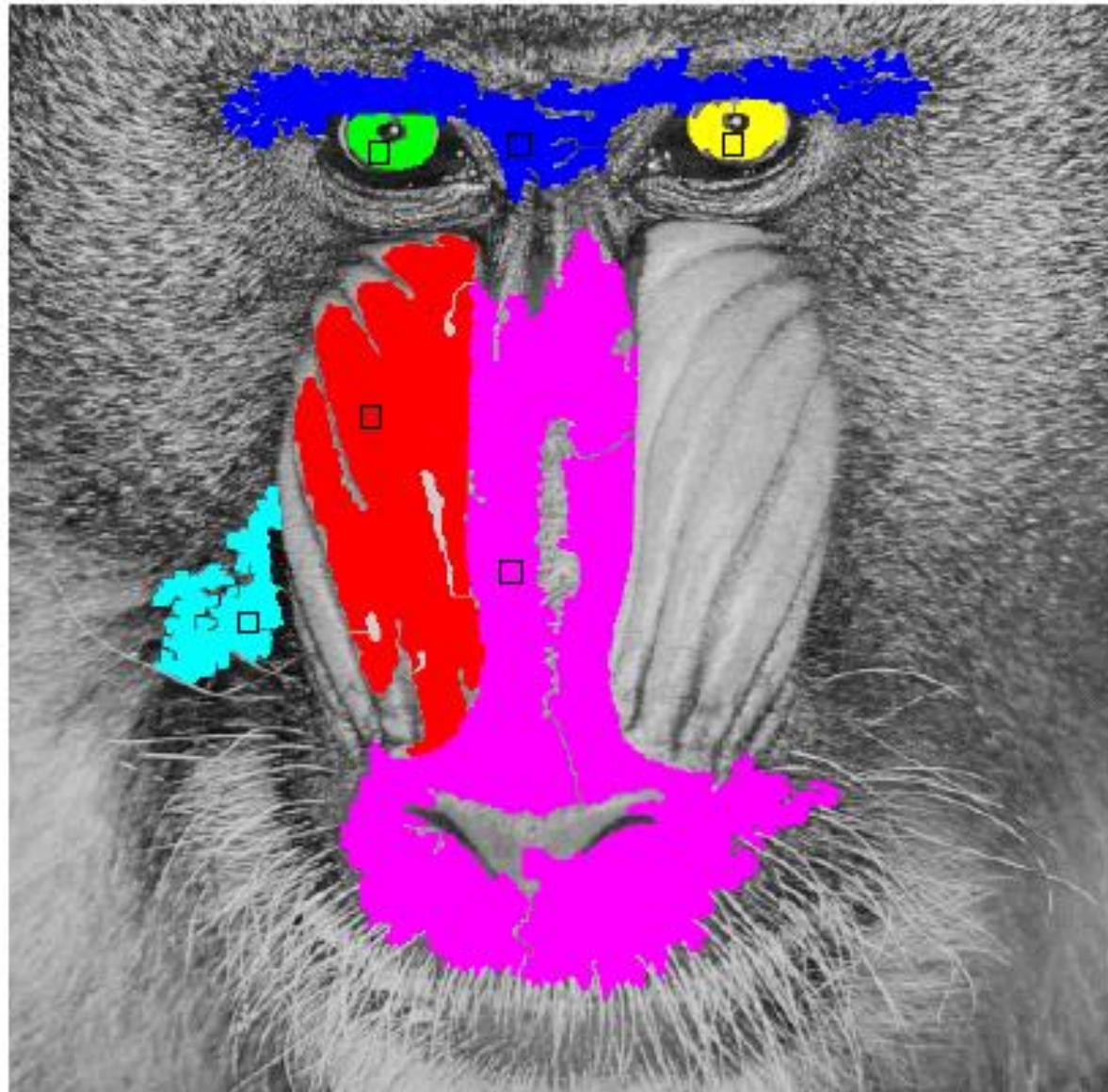
III.2 Segmentation en régions - Méthode croissance des régions

- On part d'un germe (*seed*) et on l'étend en ajoutant les pixels voisins qui satisfont le critère d'homogénéité
- Le germe peut être choisi soit par un humain, soit de manière automatique en évitant les zones de fort contraste (*gradient important*)









III.2 Segmentation en régions

Méthode croissance des régions (Region growing)

Algorithme croissance de région

Créer la liste « [S] » des points de départs (triée pour avoir le centre des plus gros blocs d'abord)

Pour chaque pixel « P » dans la liste « [S] »

Si le pixel « P » est déjà associé à une région, alors prendre
 le pixel « P » suivant dans la liste « [S] »

- Créer une nouvelle région « [R] »
- Ajouter le pixel « P » dans la région « [R] »
- Calculer la valeur/couleur moyenne de « [R] »
- Créer la liste « [m] » des pixels voisins du pixel « P »

Pour chaque pixel « P_n » dans la liste « [m] »

Si (P_n n'est pas associé à une région ET $R + P_n$ est homogène) Alors :

- Ajouter le pixel « P_n » dans la région « [R] »
- Ajouter les pixels voisins de « P_n » dans la liste « [m] »
- Recalculer la valeur/couleur moyenne de « [R] »

Fin Si

Fin Pour

Fin Si

Fin Pour

$$P(R) = \text{Vrai} \Leftrightarrow \forall p \in R, |I(p) - M_R| < \text{valeur donnée}$$

Exemple

Critère utilisé : Valeur Moyenne

10	11	12
10	11	12
9	9	12

Parcours des voisins se fait dans les directions : haut, bas,
gauche, droite

$$P(R) = \text{Vrai} \Leftrightarrow \forall p \in R, |I(p) - M_R| <= 2$$

Exemple

Critère utilisé : Valeur Moyenne

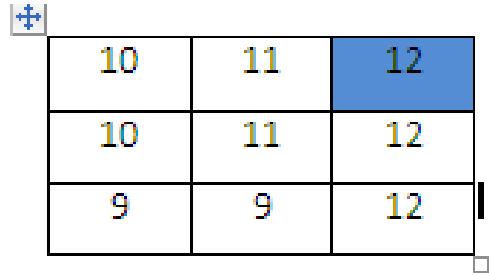
10	11	12
10	11	12
9	9	12

Parcours des voisins se fait dans les directions : haut, bas,
gauche, droite

$$P(R) = \text{Vrai} \Leftrightarrow \forall p \in R, |I(p) - M_R| <= 2$$

Exemple

Critère utilisé : Valeur Moyenne



10	11	12
10	11	12
9	9	12

Parcours des voisins se fait dans les directions : haut, bas,
gauche, droite

$$P(R) = \text{Vrai} \Leftrightarrow \forall p \in R, |I(p) - M_R| <= 2$$

Exemple

Critère utilisé : Valeur Moyenne

10	11	12
10	11	12
9	9	12

Parcours des voisins se fait dans les directions : haut, bas,
gauche, droite

⊕

10	11	12
10	11	12
9	9	12



- Evaluer le résultat d'une segmentation n'est pas facile :
 - Il dépend de l'application
 - Il dépend de ce qu'on veut
 - Il est subjectif et varie d'une personne à l'autre

Exemple-2

Parcours des voisins se fait dans les directions : haut, bas, gauche, droite

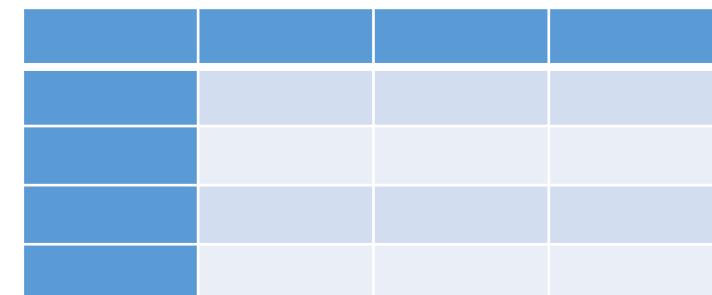
5	5	100	100
5	5	100	100
5	100	100	20
5	100	100	22
5	20	22	24

Appliquer une segmentation par croissance de régions sur cette image.

- Comme **germe** de départ que proposez vous ?
- Justifier formellement ce choix.
- Appliquer une segmentation par croissance de régions sur cette image. Détailler les résultats de la première, la seconde et la dernière itération (résultat final de la segmentation). A chaque itération, préciser le germe, son niveau de gris et ses coordonnées

Homogénéité : $\max - \min \leq 5$

première itération :
Germe de départ :
Coordonnées du germe :

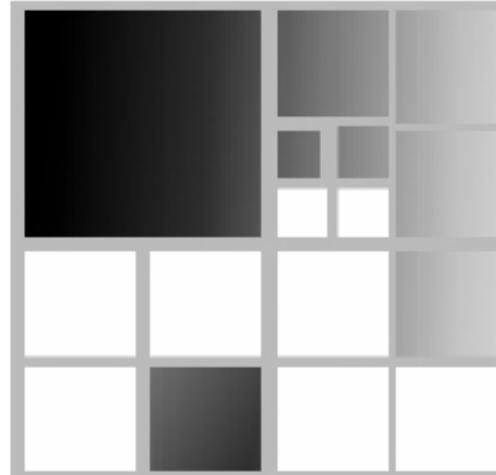


III.4 Segmentation en régions

Division et fusion (Split & Merge)

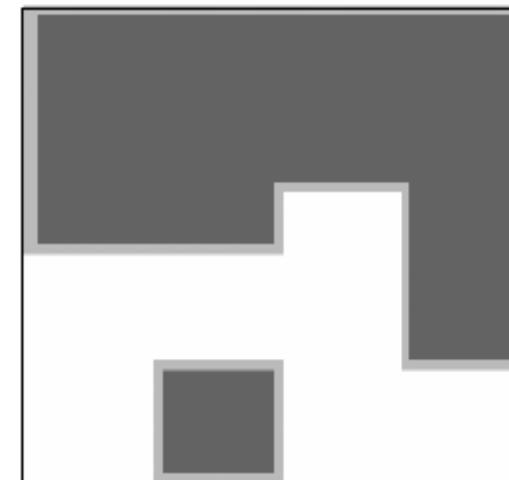
■ ***Etape de division*** (split)

- Diviser récursivement tout bloc non-homogène selon un prédicat défini
 - *variance, max-min, ...*
- La division d'un bloc donne 4 sous-blocs
- Les attributs de chaque sous-bloc sont recalculés



■ ***Etape de fusion*** (merge)

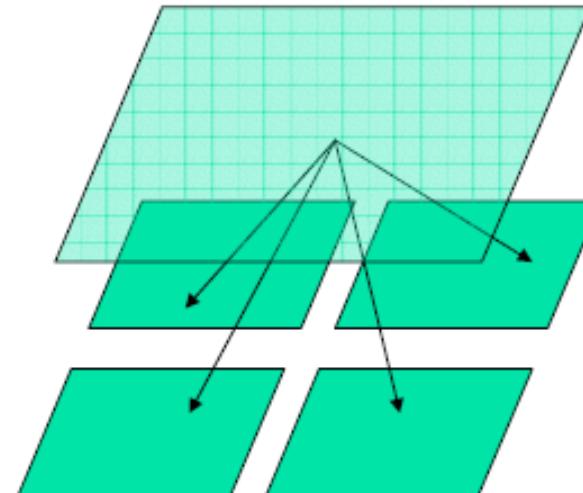
- Regrouper les blocs adjacents représentant des régions homogènes selon un prédicat défini



Etape de division

Phase 1 : Créer les zones homogènes = DIVISION (*split*)
Phase 2 : Les regrouper = FUSION (*merge*)

- L'image est stockée dans un arbre
 - Au début, arbre racine = image complète
- Récursivement, chaque feuille F est subdivisée en 4 si elle n'est pas assez homogène
 - les 4 sous-images sont ajoutées en tant que feuilles de F
- L'algorithme poursuit tant qu'il reste des feuilles non homogènes à diviser



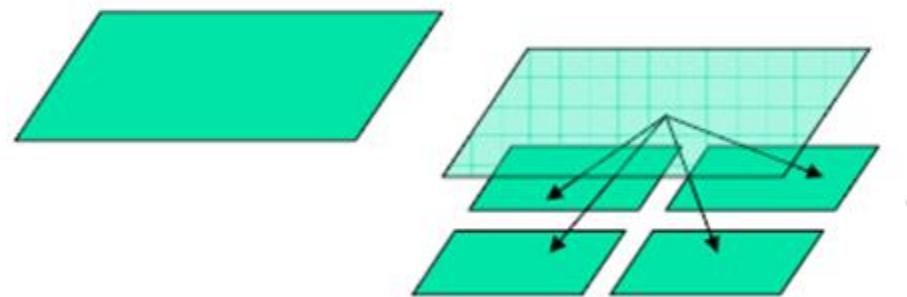
Etape de division

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Image initiale

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Division 1



*Homogénéité = critère sur la variance
(ou $\max-\min \leq 1$)*

Etape de division

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

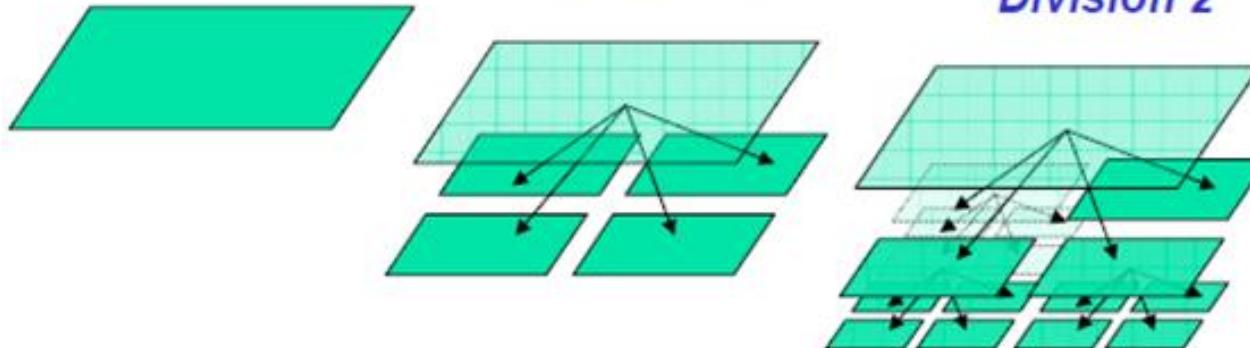
Image initiale

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Division 1

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Division 2



*Homogénéité = critère sur la variance
(ou $\max - \min \leq 1$)*

Etape de division

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Image initiale

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

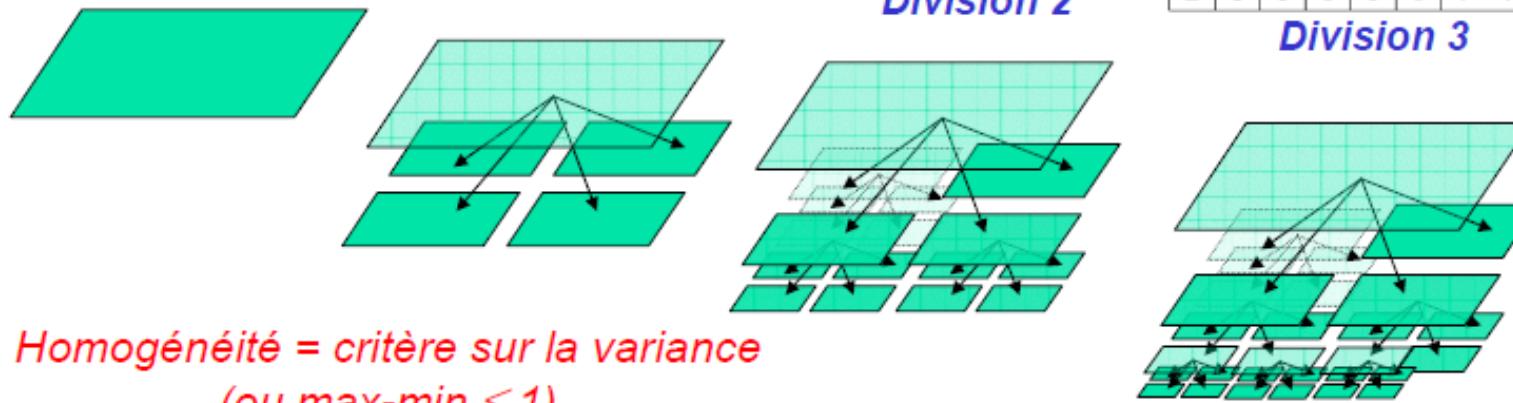
Division 1

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Division 2

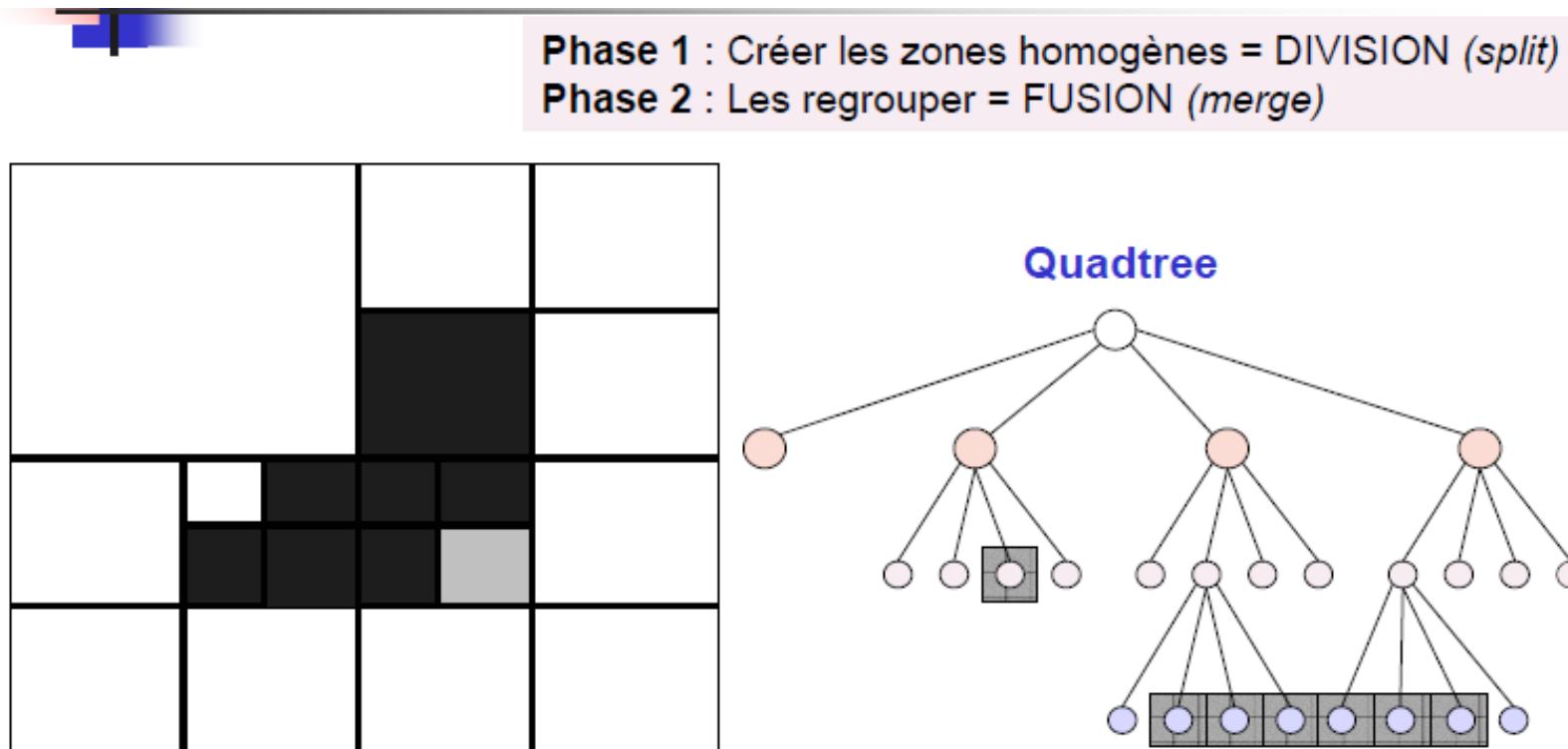
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Division 3



Etape de fusion

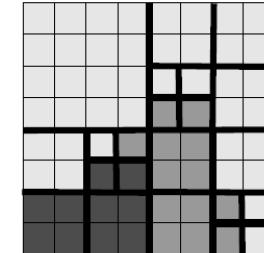
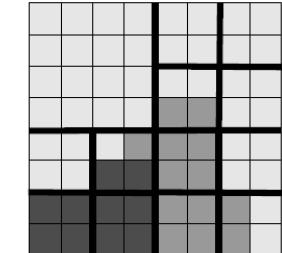
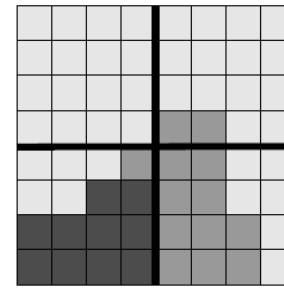
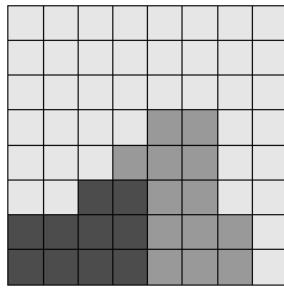
- 1- **Sur-segmentation** de l'image selon un prédicat très strict
- 2- Regroupement des régions adjacentes selon **un prédicat plus lâche**



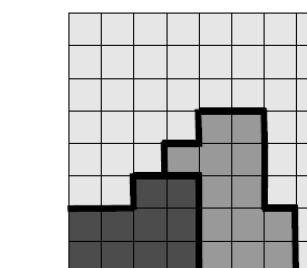
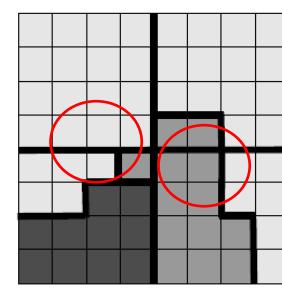
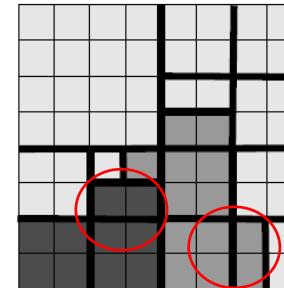
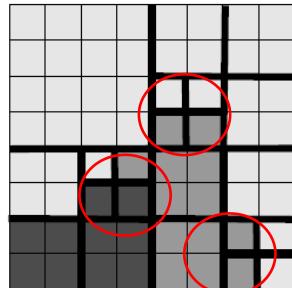
Connecter les régions adjacentes homogènes

Illustration 2 : Division ensuite fusion

Division



Fusion





Lenna



Après division



Après fusion



après division



après fusion

Split & Merge : conclusion

- Split : la géométrie du découpage a une grande influence sur le résultat de la segmentation. Par exemple, le split en quadtree fait apparaître des régions carrées.
- Merge : l'ordre dans lequel est réalisé le regroupement des régions a aussi une influence sur le résultat. Comme illustré dans les figures précédentes, les algorithmes commencent en général à fusionner les régions les plus petites.

Split & Merge : conclusion

- Aucune méthode ne fonctionne pour toutes les images
- Pas de garantie, pas de recette miracle !
- Comment choisir convenablement les seeds de chaque région ?
- En général, l'ordre dans lequel les régions sont construites, mais aussi l'ordre dans lequel sont ajoutés les pixels dans une région a une grande influence sur le résultat.

Partie 2

Vision 3D

PLAN

- 1) Vision 3D : Introduction**
- 2) Représentation surfacique des objets 3D-Maillage Polygonal**
- 3) Reconstruction 3D**
- 4) Visualisation 3D**



Vision 3D

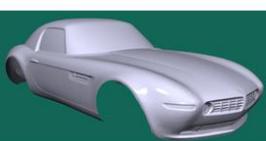
Introduction

- La vision 2D se limite à la perception de la longueur et de la largeur des objets sur une surface plane.
- La vision 3D, ou vision tridimensionnelle, fait référence à la capacité de percevoir les objets dans l'espace en trois dimensions : longueur, largeur et profondeur.
- La vision 3D permet de percevoir la profondeur et la distance entre les objets.

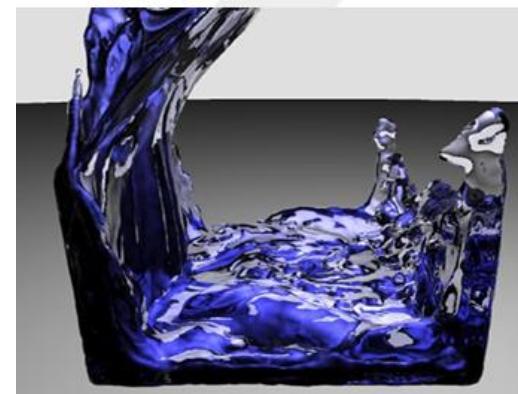
Vision 3D

Introduction

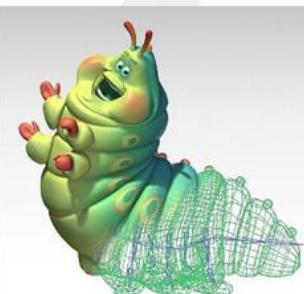
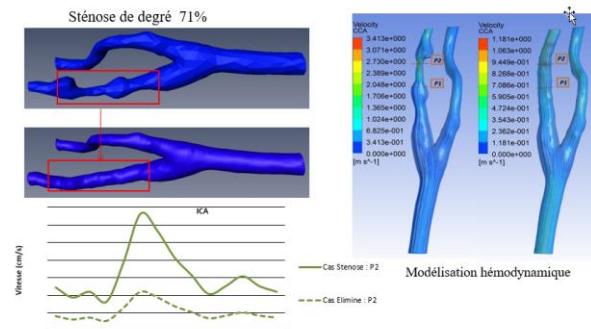
La vision 3D est essentielle dans de nombreux domaines, tels que la perception spatiale, la navigation, la réalité virtuelle, la robotique, l'imagerie médicale, le cinéma 3D et bien d'autres encore.



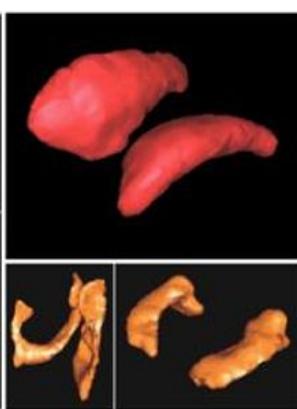
Construction Design : Maquette



- Modèles en entrées
- Mesures pour les modèles physiques



Modélisation de formes libres

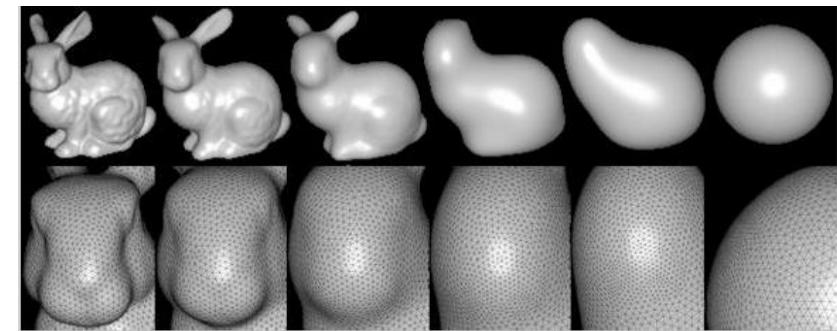
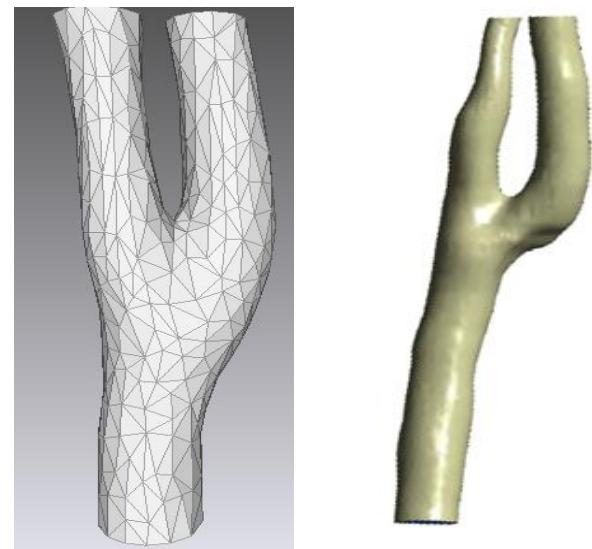
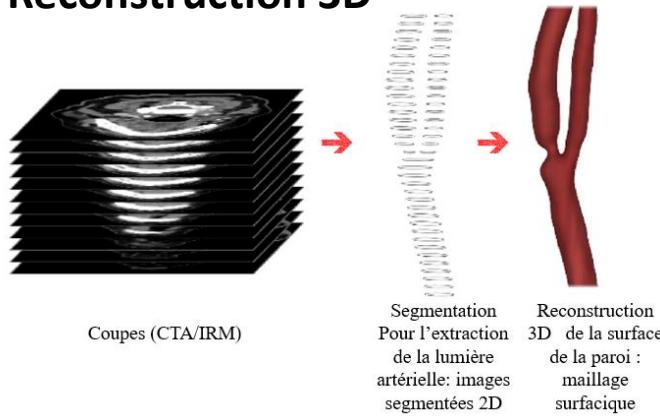


- Domaine Médical**
- Visualisation de données issues des scanner (surfacique/volumique)
 - Reconstruction 3D des organes
 - Simulation des déformations (opérations virtuelle)

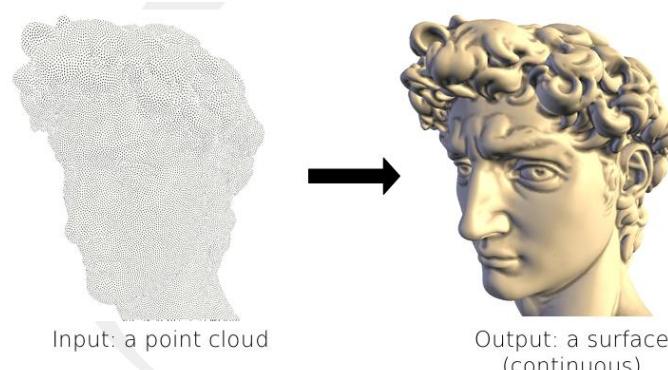
Vision 3D

Introduction

Reconstruction 3D



Reconstruction de surface à partir d'images 2D



Rendu des surfaces reconstruites en 3D

Reconstruction de surface à partir de nuage de points

Simplification, lissage et remaillage des surfaces

Vision 3D

Modélisation des objets 3D

- Comment représenter des objets numériques ?
- Comment les créer/manipuler/visualiser/ les stocker sur machine ?
- **Modélisation 3D** : vise la création des objets en trois dimension, à partir ou non d'objets réels.
- **Modèle d'un objet** : une représentation simplifiée d'un objet qui permet de le visualiser plus aisément.
- **Intérêt** : la construction d'un modèle pour représenter la structure géométrique d'un objet est intéressante car certaines caractéristiques du modèle peuvent être étudiées plus facilement que celles de l'objet lui-même.

Vision 3D

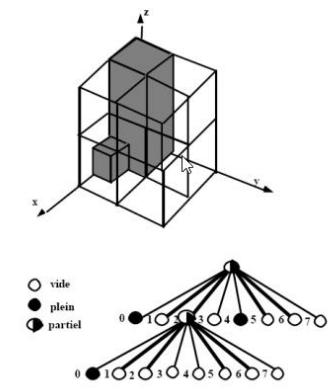
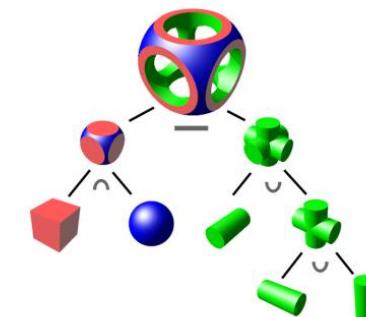
Modélisation des objets 3D

- La représentation des objets en 3D peut être effectuée par différentes méthodes, en fonction du contexte et des besoins spécifiques.
- Un objet 3D peut être représenté soit sous forme de **volume** soit sous forme de **surface**

Exemple :

❖ Modèles volumiques (volumetric) :

- Décrivent la forme d'un objet en termes de volume.
- Par exemple : des objets simples peuvent être représentés à l'aide de primitives (quadriques, polyèdres, sphère, etc.).
- Des objets plus complexes sont construits par combinaison des objets simples par utilisation d'opérations ensemblistes.

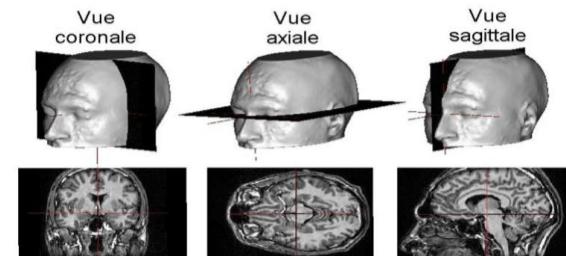
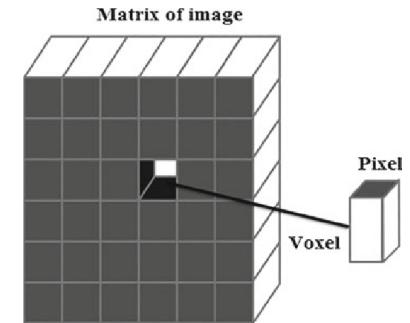


encodage des solides par des octrees.

Vision 3D

Modélisation des objets 3D

- ❖ **Représentations basées sur les voxels (volumetric pixels) :** Les voxels sont des éléments volumétriques qui forment une grille dans l'espace 3D.
- ❖ Ils sont souvent utilisés pour représenter des volumes et sont fréquemment utilisés dans les domaines tels que l'imagerie médicale et la simulation numérique.



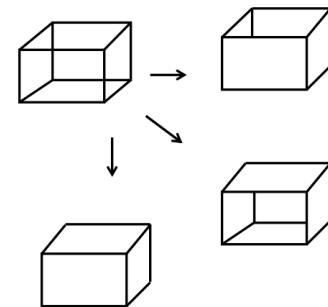
Vision 3D

Modélisation des objets 3D

❖ **NUAGE DE POINTS** : Cette représentation consiste en un ensemble de points dans l'espace 3D, où chaque point représente un point de surface de l'objet.

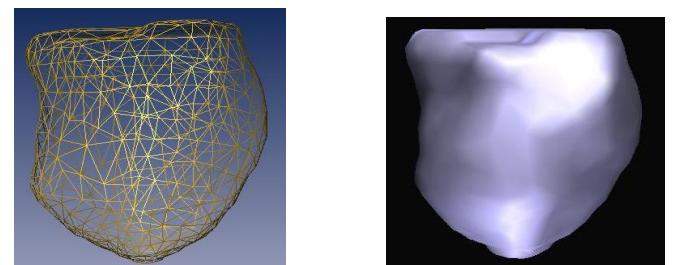


❖ **Modèles filaires (wireframe)** : Les objets sont représentés à partir d'un ensemble de segments de droite reliés éventuellement par leurs extrémités. Il permet d'avoir une représentation géométrique globale de l'objet.



❖ **Polygones (Mesh)** : Cette méthode divise la surface de l'objet en petits polygones pour former une maille (mesh).

Les modèles de polygones sont très courants dans les applications de modélisation 3D et de graphiques par ordinateur en raison de leur simplicité et de leur efficacité.



Représentation par maillage triangulaire et rendu surfacique

Vision 3D

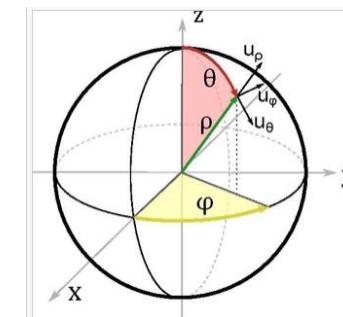
Modélisation des objets 3D

- ❖ **Représentation paramétrique de surface** : Dans cette méthode, la surface est définie par une ou plusieurs fonctions paramétriques qui prennent des coordonnées (u, v) en entrée et produisent les coordonnées (x, y, z) correspondantes de la surface en sortie.
- ❖ Les surfaces paramétriques sont souvent utilisées dans la modélisation mathématique et la conception assistée par ordinateur, ...

Une surface paramétrique dans l'espace \mathbb{R}^3 est définie par une fonction f :

$$f: D \times E \rightarrow \mathbb{R}^3$$

$$u, v \rightarrow p(u, v) = \begin{cases} x(u, v) = f_x(u, v) \\ y(u, v) = f_y(u, v) \\ z(u, v) = f_z(u, v) \end{cases}$$



$$\begin{cases} x = \rho \sin \theta \cos \varphi \\ y = \rho \sin \theta \sin \varphi \\ z = \rho \cos \theta \end{cases}$$

- ❖ Etc.

Exemple de représentation explicite, cas de la sphère

Vision 3D

Modélisation des objets 3D

- ❑ Plusieurs techniques sont utilisées pour la représentation des surfaces en 3D.
- ❑ Le profil extérieur des objets 3D est représenté par une surface.
- ❑ Dans la suite, nous nous focalisons sur **la modélisation géométrique des surfaces** et plus précisément par **maillages polygonaux**

Vision 3D

Représentation surfacique des objets 3D

Afin de manipuler des objets 3D et implémenter des algorithmes de reconstruction, de visualisation 3D de calcul de mesures, il est important de définir d'abord les **concepts de base de la 3D**, des **objets 3D**, la manière avec laquelle un objet 3D est **représenté** et **stocké** sur machine, et les **principaux formats d'objet** 3D.

Concepts de base et notions du 3D:

□ Dimension d'un objet (espace topologique)

- Point : 0D
- Droite, Segment, Courbe : 1D
- Plan, Surface : 2D
- Volume : 3D

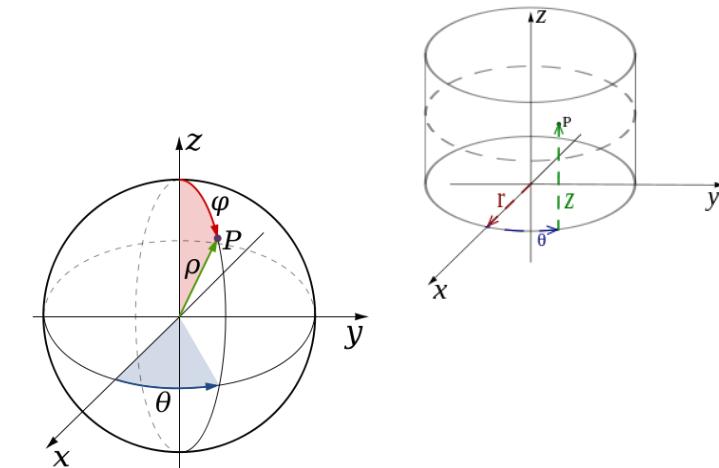
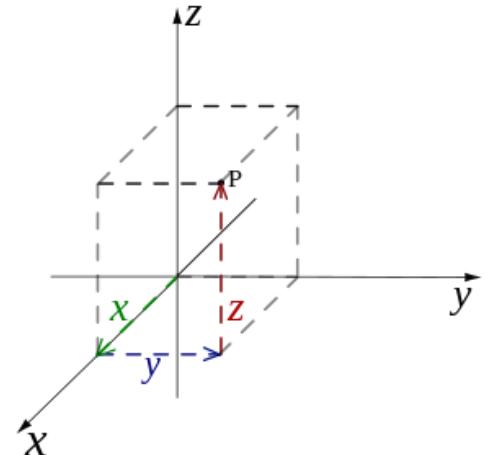
Vision 3D

Représentation surfacique des objets 3D

Concepts de base et notions du 3D:

A) Géométrie

- ❖ L'information sur la géométrie est rapportée par les **coordonnées tridimensionnelles (x, y, z) d'un point (vertex, vertices) de l'espace 3D.**
- ❖ On utilise souvent le repère cartésien où le point est défini par ses coordonnées x,y et z.
- ❖ Il existe d'autres systèmes de coordonnées permettant de repérer un point dans le plan ou dans l'espace : les coordonnées sphériques et les coordonnées cylindriques

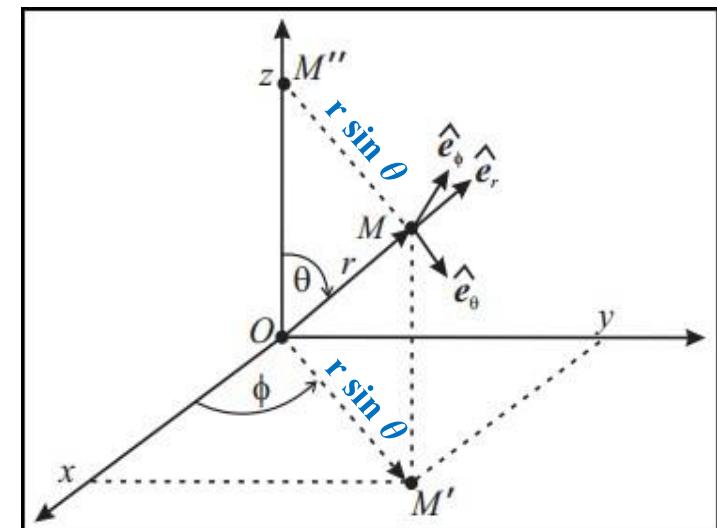


Vision 3D

Représentation surfacique des objets 3D

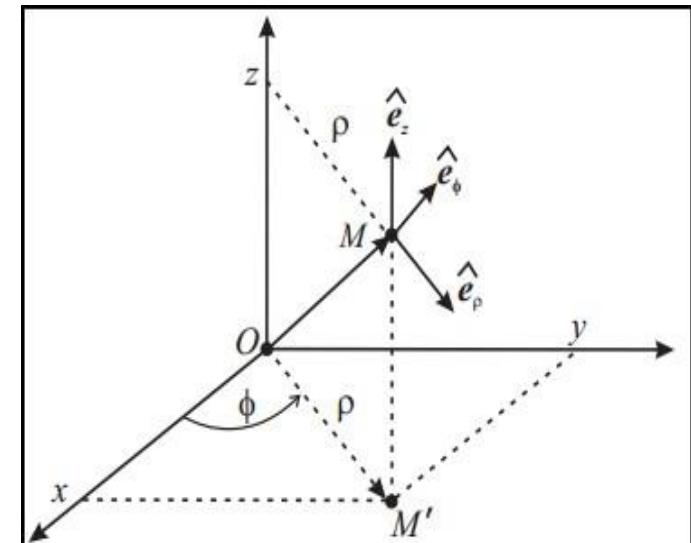
- ❖ Les coordonnées cartésiennes d'un point M (r) s'écrivent en fonction des coordonnées sphériques comme suit :

$$x = r \sin \theta \cos \phi, y = r \sin \theta \sin \phi, z = r \cos \theta$$



- ❖ Les coordonnées cartésiennes d'un point M (r) s'écrivent en fonction des coordonnées cylindriques comme suit :

$$x = \rho \cos \phi, y = \rho \sin \phi, z = z$$



Vision 3D

Représentation surfacique des objets 3D

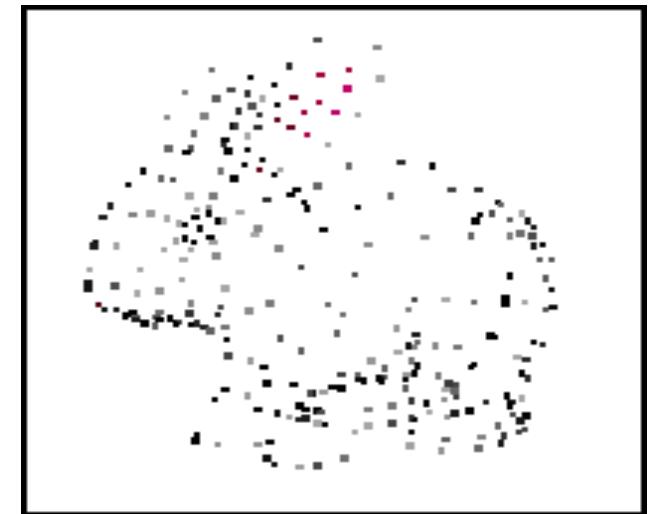
Concepts de base et notions du 3D:

B) Nuages de points 3D

□ L'information sur la **géométrie** est rapportée par les coordonnées tridimensionnelles (x, y, z) d'un point de l'espace 3D.

□ Nous désignons par nuage de points 3D un ensemble de points 3D P_i , où $i = 1, 2, \dots, N$.

Chaque point P_i du nuage N est décrit par ses coordonnées respectives (X_i, Y_i, Z_i).



Vision 3D

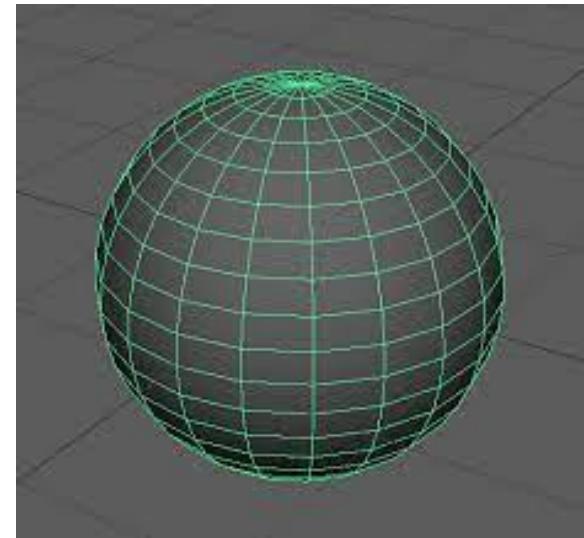
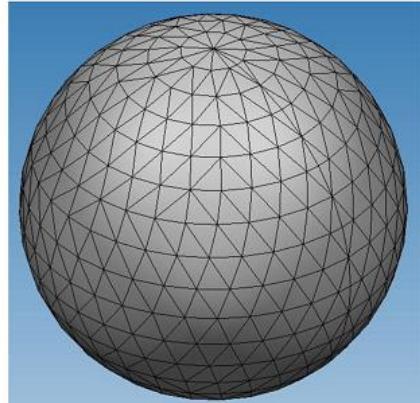
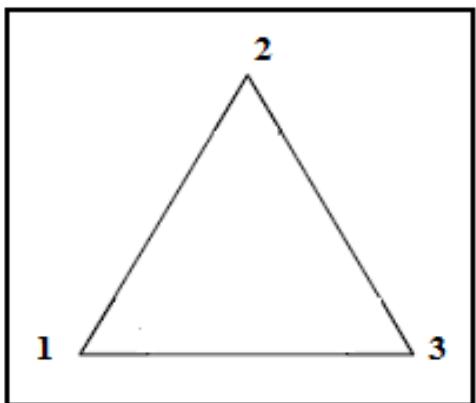
Représentation surfacique des objets 3D

Concepts de base et notions du 3D:

C) Facette

La facette (élément de surface) est définie par une liste ordonnée et orientée de sommets.

En particulier, une facette triangulaire se compose de 3 sommets liés par 3 arêtes.



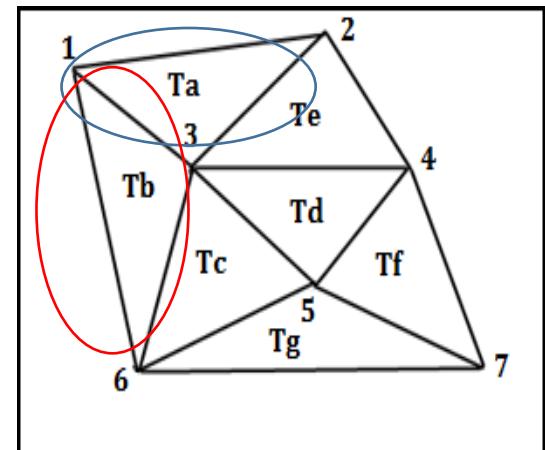
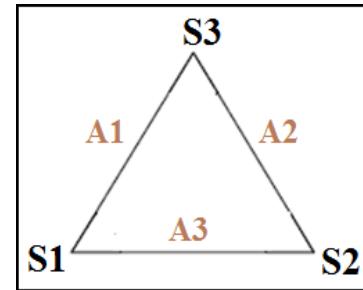
Vision 3D

Représentation surfacique des objets 3D

Concepts de base et notions du 3D:

D) Topologie et Facette

- ❖ La topologie décrit la **relation d'adjacence** entre les sommets traduisant la manière dont ils sont connectés entre eux.
- ❖ La facette (élément de surface) ou primitive est définie par une liste ordonnée et orientée de sommets.
- ❖ Par exemple **T_a** est le triangle formé par la succession des sommets 1, 2, 3 (topologie) où chaque sommet est défini par ses coordonnées x, y et z (géométrie).
 - Les voisins du sommet 1 sont les sommets 2 et 3, ils forment le triangle **T_a**
 - Les sommets 1, 3 et 6 forment le triangle **T_b**



Vision 3D

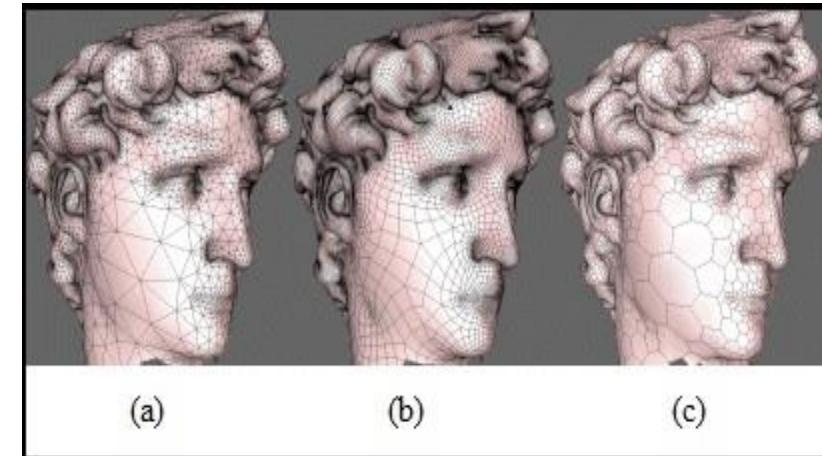
Représentation surfacique des objets 3D

Maillage Polygonal

Dans le cas de surface, on parle de maillage polygonal : **une collection de sommets, côtés et polygone (face plane)**. Plus précisément :

- les sommets (en anglais, vertex ou vertices) sont représentés par des points du plan ou de l'espace, et peuvent être munis de propriétés additionnelles : couleur, normales pour l'affichage, par exemple ;
- les arêtes (côté, en anglais edge) sont des connexions entre deux sommets, et sont situés sur la frontière d'une face ;
- les faces (facette, en anglais, face) sont des polygones du plan ou de l'espace, dont les sommets et les arêtes appartiennent au maillage.

Les faces peuvent avoir des propriétés telles : couleur, transparence pour l'affichage.



Exemple de maillages polygonaux:
(a) facette triangulaire,
(b) facette rectangulaire,
(c) facette hexagonale.

Vision 3D

Représentation surfacique des objets 3D

Maillage Polygonal

Plus formellement, un maillage M est défini par le triplet

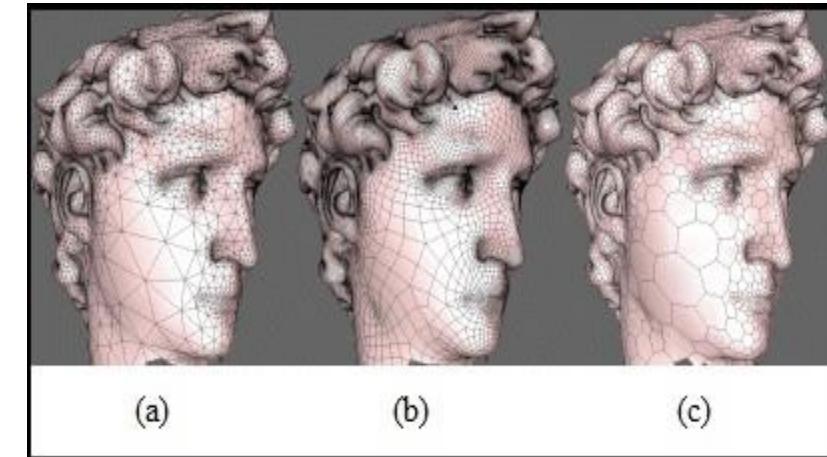
V : sommets,

E : arêtes

et F : facettes

M=triplet V, E, F.

- ❖ Un polygone est un ensemble de sommets liés par des arêtes. Une arête est partagée par deux polygones au plus.
- ❖ Un maillage polygonal est la partition en éléments polygonaux (triangles, hexagones, carré ...) pour produire une représentation approchée de la forme.



Exemple de maillages polygonaux:

- (a) facette triangulaire,
- (b) facette rectangulaire,
- (c) facette hexagonale.

Vision 3D

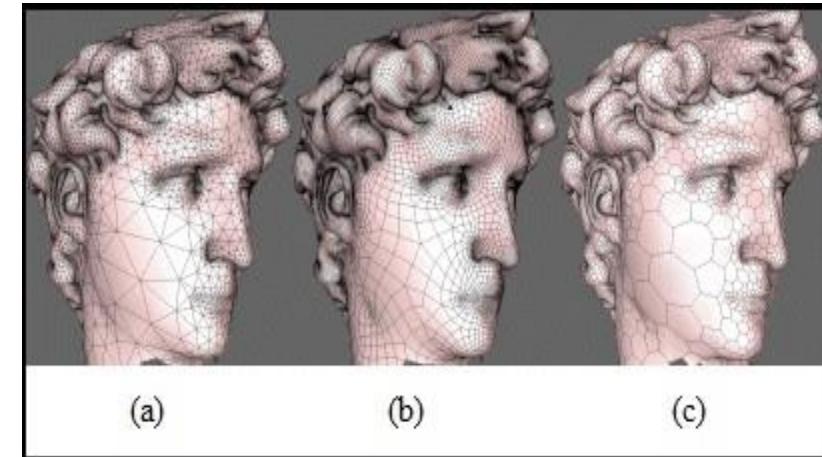
Représentation surfacique des objets 3D

Maillage Polygonal

Plus formellement, un maillage M est défini par le triplet

V : sommets,
E : arêtes
et F : facettes
M=triplet V, E, F.

- ❖ Graphe planaire : peut être représenté dans le plan sans qu'aucune arête n'en croise une autre.
- ❖ Formule d'Euler : $N_V - N_E + N_F = 2$
où N_V : nombre de sommets, N_E le nombre d'arêtes et N_F le nombre de faces du graphe.



Exemple de maillages polygonaux:
(a) facette triangulaire,
(b) facette rectangulaire,
(c) facette hexagonale.

Vision 3D

Représentation surfacique des objets 3D

Maillage Polygonal

Plus formellement, un maillage M est défini par le triplet

V : sommets, E : arêtes, et F : facettes , M=triplet V, E, F.

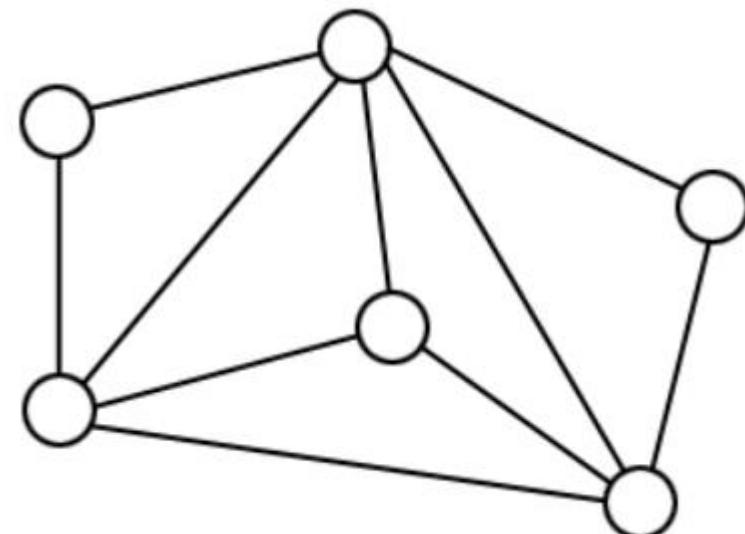
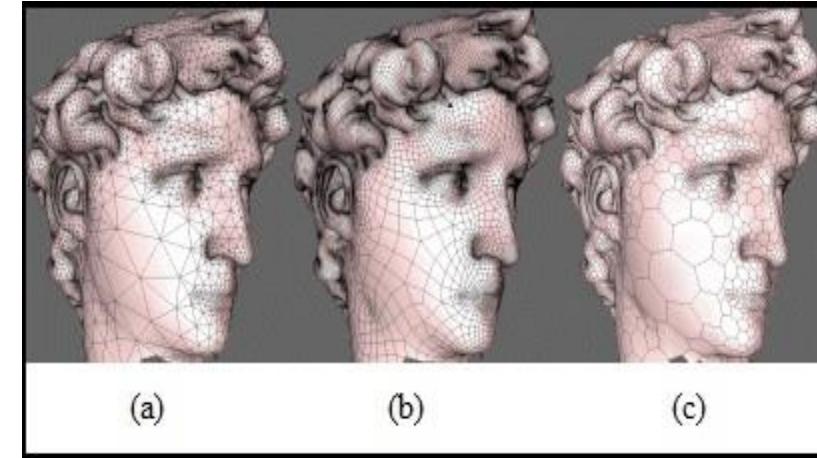
❖ Formule d'Euler : $N_V - N_E + N_F = 2$

où N_V : nombre de sommets, N_E , le nombre d'arêtes et N_F le nombre de faces du graphe.

❖ $N_V = 6, N_E = 9, N_F = 5$

❖ $N_V - N_E + N_F = 2 ?$

→ $6 - 9 + 5 = 2$

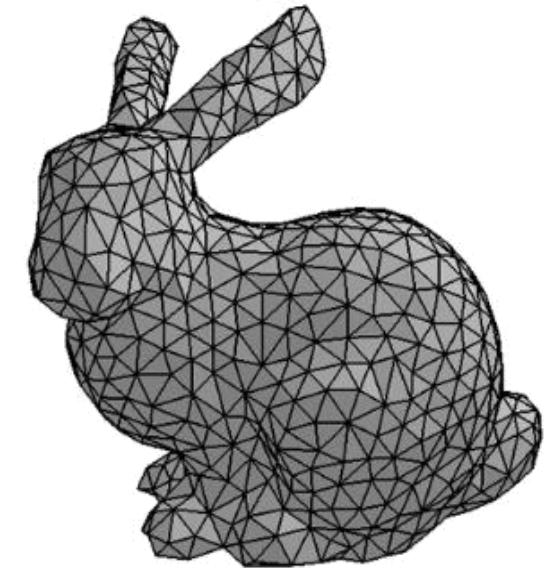


Vision 3D

Représentation surfacique des objets 3D

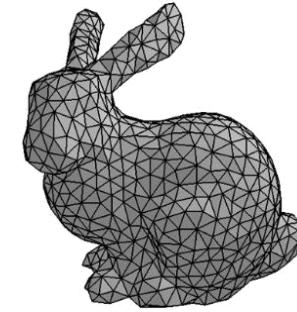
Maillage Triangulaire

- ❖ Lorsque le nombre de sommets est égal à trois, on parle de maillage triangulaire.
- ❖ La facette dans ce cas est un triangle.
- ❖ Les maillages triangulaires sont très utilisés du fait de leur simplicité et la rapidité de calcul comme par exemple dans la visualisation 3D.



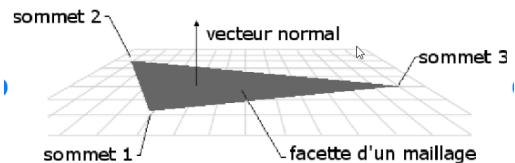
Vision 3D

Représentation surfacique des objets 3D



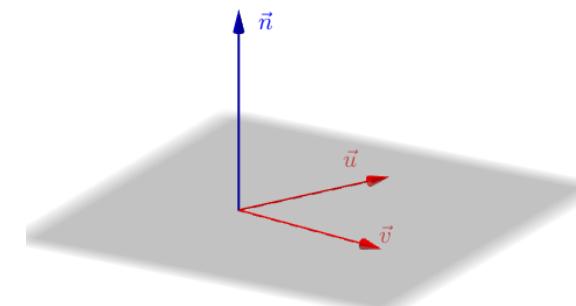
Maillage Triangulaire

- ❖ Les maillages triangulaires sont très utilisés du fait de leur simplicité et la rapidité de calcul comme par exemple dans la visualisation 3D.
- ❖ Vecteur normal à un plan



Définition : Un vecteur non nul \vec{n} de l'espace est normal à un plan P lorsqu'il est orthogonal à tout vecteur admettant un représentant dans P .

Théorème : Un vecteur non nul \vec{n} de l'espace est normal à un plan P s'il est orthogonal à deux vecteurs non colinéaires de P .



Vision 3D

Exemple de calcul du Vecteur normal à un plan

Dans un repère orthonormé, soit $A\begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix}$, $B\begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix}$ et $C\begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix}$.

Déterminer un vecteur normal au plan (ABC).

On a : $\overrightarrow{AB}\begin{pmatrix} -2 \\ 1 \\ 3 \end{pmatrix}$ et $\overrightarrow{AC}\begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$.

Soit un vecteur $\vec{n}\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ orthogonal au plan (ABC). Il est tel que

$$\begin{cases} \vec{n} \cdot \overrightarrow{AB} = 0 \\ \vec{n} \cdot \overrightarrow{AC} = 0 \end{cases} \text{ Soit } \begin{cases} -2a + b + 3c = 0 \\ a - 2b = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} -2 \times 2b + b + 3c = 0 \\ a = 2b \end{cases}$$

$$\Leftrightarrow \begin{cases} -3b + 3c = 0 \\ a = 2b \end{cases}$$

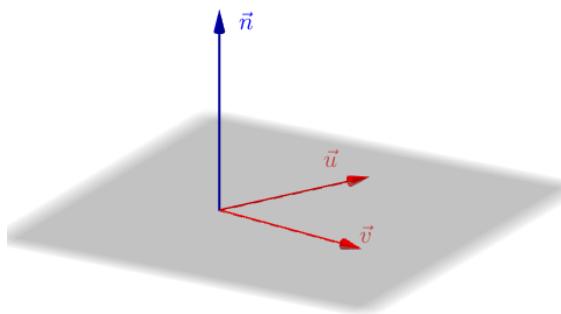
$$\Leftrightarrow \begin{cases} c = b \\ a = 2b \end{cases}$$

Prenons par exemple, $b = 1$ alors $c = 1$ et $a = 2$.

Le vecteur $\vec{n}\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$ est donc normal au plan (ABC).

Définition : Un vecteur non nul \vec{n} de l'espace est normal à un plan P lorsqu'il est orthogonal à tout vecteur admettant un représentant dans P .

Théorème : Un vecteur non nul \vec{n} de l'espace est normal à un plan P s'il est orthogonal à deux vecteurs non colinéaires de P .



Vision 3D

Exemple de calcul du Vecteur normal à un plan

Dans un repère orthonormé, soit $A \begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix}$, $B \begin{pmatrix} -1 \\ 3 \\ 1 \end{pmatrix}$ et $C \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix}$.

Déterminer un vecteur normal au plan (ABC).

On a : $\overrightarrow{AB} \begin{pmatrix} -2 \\ 1 \\ 3 \end{pmatrix}$ et $\overrightarrow{AC} \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$.

$$\vec{n} = \overrightarrow{AB} \times \overrightarrow{AC}$$

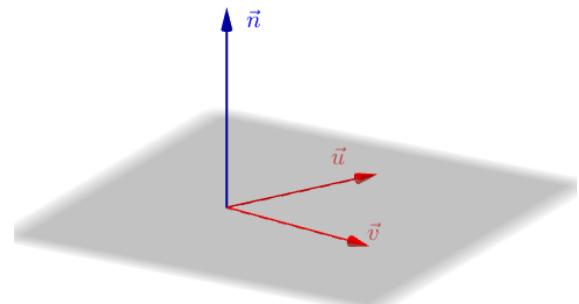
$$= \begin{vmatrix} i & j & k \\ -2 & 1 & 3 \\ 1 & -2 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 3 \\ -2 & 0 \end{vmatrix} \vec{i} - \begin{vmatrix} -2 & 3 \\ 1 & 0 \end{vmatrix} \vec{j} + \begin{vmatrix} -2 & 1 \\ 1 & -2 \end{vmatrix} \vec{k}$$

$$= 6 \vec{i} + 3 \vec{j} + 3 \vec{k}$$

$$\vec{n} = (2, 1, 1)$$

Définition : Un vecteur non nul \vec{n} de l'espace est normal à un plan P lorsqu'il est orthogonal à tout vecteur admettant un représentant dans P .

Théorème : Un vecteur non nul \vec{n} de l'espace est normal à un plan P s'il est orthogonal à deux vecteurs non colinéaires de P .



Vision 3D

Exemple de calcul du Vecteur normal à un plan

Soient les deux vecteurs $(9, 7, -8)$ et $(-2, 2, -1)$

On peut calculer le produit vectoriel entre ces deux vecteurs

Définition : Un vecteur non nul \vec{n} de l'espace est normal à un plan P lorsqu'il est orthogonal à tout vecteur admettant un représentant dans P .

Théorème : Un vecteur non nul \vec{n} de l'espace est normal à un plan P s'il est orthogonal à deux vecteurs non colinéaires de P .

$$\vec{n} = (9, 7, -8) \times (-2, 2, -1)$$

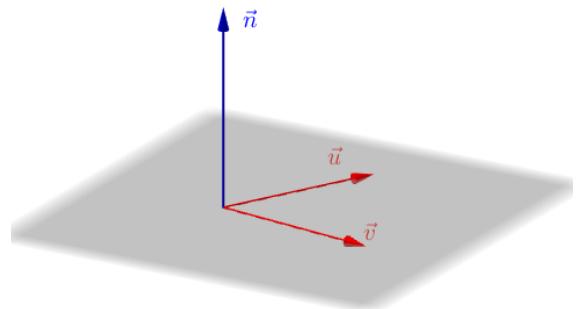
$$= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 9 & 7 & -8 \\ -2 & 2 & -1 \end{vmatrix}$$

$$= \begin{vmatrix} 7 & -8 \end{vmatrix} \vec{i} - \begin{vmatrix} 9 & -8 \end{vmatrix} \vec{j} + \begin{vmatrix} 9 & 7 \end{vmatrix} \vec{k}$$

$$= (7 \times (-1) - (-8) \times 2) \vec{i} - (9 \times (-1) - (-8) \times (-2)) \vec{j} + (9 \times 2 - 7 \times (-2)) \vec{k}$$

$$= 9\vec{i} + 25\vec{j} + 32\vec{k}$$

$$= (9, 25, 32).$$



Vision 3D

Représentation surfacique des objets 3D

Quelle représentation choisir pour stocker le maillage polygonal ?

II. Représentation des maillages triangulaires

Pour stocker un maillage polygonal sur l'ordinateur, il existe principalement **trois types** de représentations:

Explicite, avec des pointeurs sur une liste d'arêtes et avec des pointeurs vers une liste de sommets.

◊ **Représentation Explicite**

Chaque polygone représenté par

$$P = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n))$$

- Sommets stockés dans l'ordre d'apparition en tournant autour du polygone.
- Il y a des arêtes entre chacun des sommets successifs dans la liste et entre le dernier et le premier.

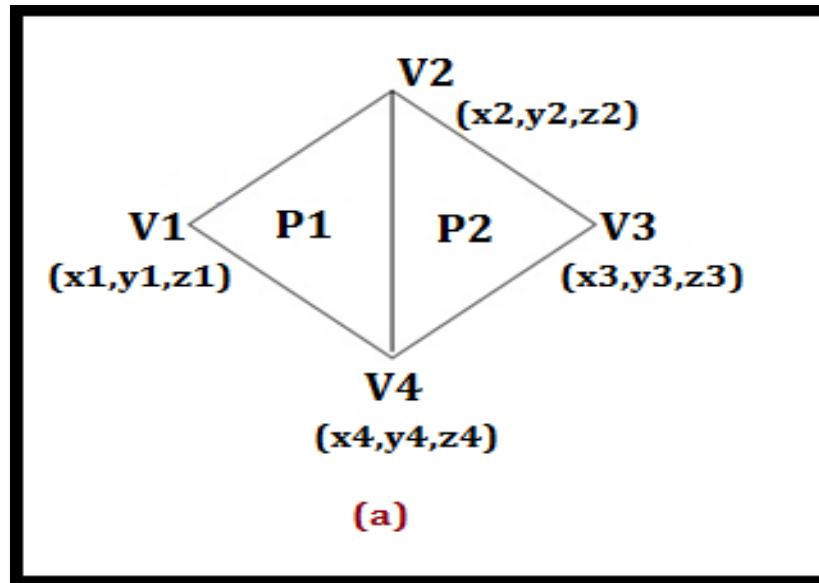
Inconvénients : espace perdu, duplication des coordonnées des sommets partagés.

Vision 3D

Représentation surfacique des objets 3D

Exemple de représentation explicite

- $P1 = ((x1, y1, z1), (x2, y2, z2), (x4, y4, z4))$
- $P2 = ((x4, y4, z4), (x2, y2, z2), (x3, y3, z3))$



Vision 3D

Représentation surfacique des objets 3D

II. Représentation des maillages triangulaires

- ◊ **Avec des pointeurs vers une liste d'arêtes**

- ✓ Liste de sommets V.
 - ✓ Polygone=liste de pointeurs sur une liste d'arêtes,
 - ✓ chaque arête n'apparaît qu'une fois.
 - ✓ Chaque arête pointe sur deux sommets la définissant dans la liste de sommets et sur le ou les deux polygones à laquelle elle appartient.
-
- $P = (E_1, E_2, \dots, E_n)$
 - Et une arête commune $E = (V_1, V_2, P_1, P_2)$

Vision 3D

Représentation surfacique des objets 3D

Avec des pointeurs vers une liste d'arêtes

$V=((x_1,y_1,z_1),(x_2,y_2,z_2),(x_3,y_3,z_3),(x_4,y_4,z_4))$

$E1= (V1, V2, P1, \lambda)$

$E2= (V2, V3, P2, \lambda)$

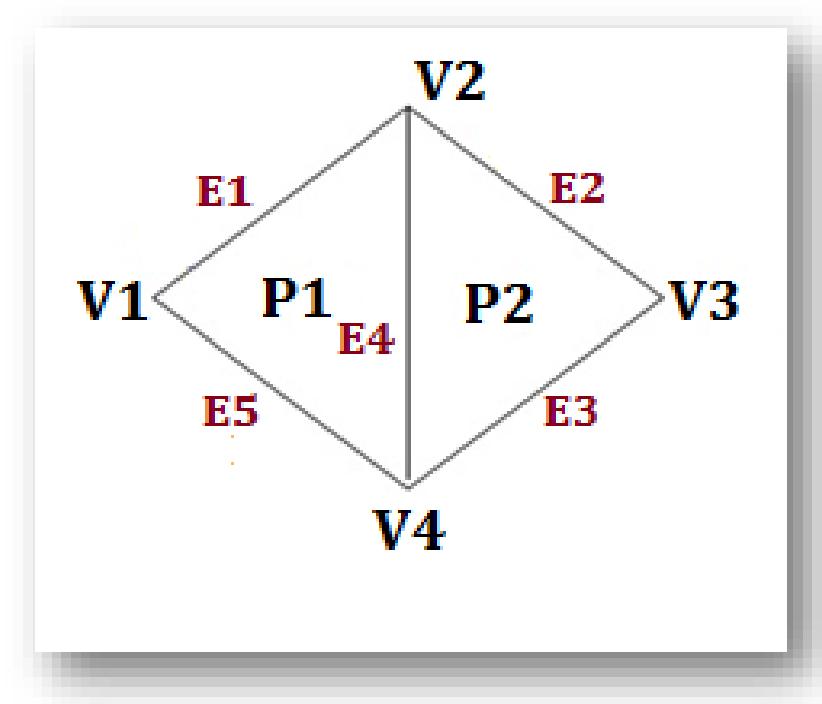
$E3= (V3, V4, P2, \lambda)$

$E4= (V2, V4, P1, P2)$

$E5= (V4, V1, P1, \lambda)$

$P1= (E1, E4, E5)$

$P2= (E2, E3, E4)$



Quand une arête n'appartient qu'un à un polygone P1 ou P2, lamda est nul

Vision 3D

Représentation surfacique des objets 3D

II. Représentation des maillages triangulaires

◊ **Avec des pointeurs vers une liste de sommets**

- ✓ Chaque sommet est stocké une seule fois dans la liste des sommets.
 $V = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots (x_n, y_n, z_n))$
- ✓ Un polygone est défini par une liste d'indices (ou de pointeurs) dans la liste des sommets.

Avantage :

- Chaque sommet est présent une fois dans la liste des sommets.
- Economie de place.
- Possibilité de changer facilement les coordonnées d'un sommet.

Vision 3D

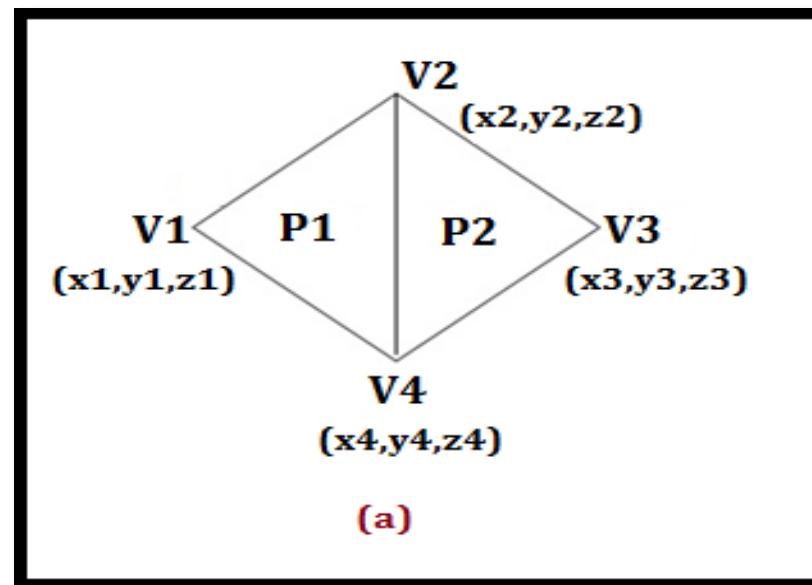
Représentation surfacique des objets 3D

Exemple de représentation **Avec des pointeurs vers une liste de sommets**

$$V = ((x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4))$$

$$P1 = (1, 2, 4)$$

$$P2 = (4, 2, 3)$$

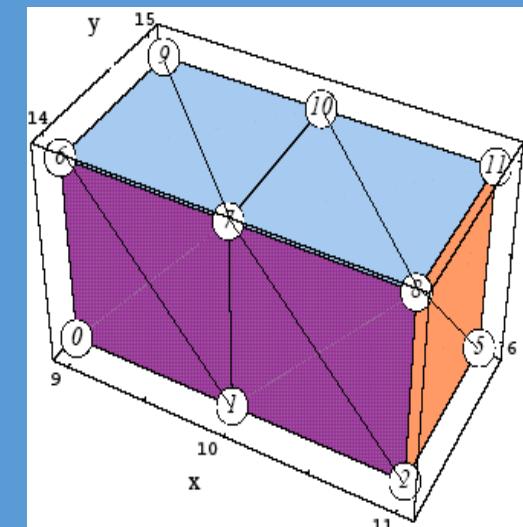


Vision 3D

Représentation surfacique des objets 3D

Maillage triangulaire : Exemple de représentation **Avec des pointeurs vers une liste de sommets**

Numéro sommet	x	y	z	Facettes
0	{9}	14	6}	{0, 1, 6}
1	{10}	14	6}	{1, 7, 6}
2	{11}	14	6}	{1, 2, 7}
3	{9}	14	6}	{2, 8, 7}
4	{10}	14	6}	{9, 6, 7}
4	{11}	14	6}	{9, 7, 10}
6	{9}	14	7	{7, 8, 10}
7	{10}	14	7	{8, 11, 10}
8	{11}	14	7	{9, 3, 0}
9	{9}	14	7	{0, 6, 9}
10	{10}	14	7	{4, 4, 11}
11	{11}	14	7	{4, 10, 11}
				{4, 3, 10}
				{3, 9, 10}
				{0, 3, 4}
				{0, 4, 1}
				{1, 4, 6}
				{1, 4, 2}
				{8, 2, 4}
				{8, 4, 11}



Vision 3D

Représentation surfacique des objets 3D

III. Formats de fichiers 3D

- ❖ Il existe des dizaines de formats de fichiers 3D, par exemple .obj, .3ds, .dxf,..
- ❖ Le format .obj. est un format reconnu par sa popularité et sa simplicité.
- ❖ Il utilise notamment la représentation d'un polygone maillé avec des pointeurs vers une liste détaillée de sommets.

- ❖ Le fichier `obj' a la structure suivante :
 - Une entête et deux blocs;
 - Le premier BLOC est composé par :
 - ✓ des lignes qui sont précédées par des "v" pour indiquer qu'il s'agit d'un sommet (vertices).
 - ✓ Chaque sommet est décrit par ses coordonnées(x y z).
 - Le deuxième bloc est composé par :
 - ✓ des lignes qui sont précédées par des "f" pour indiquer qu'il s'agit d'une facette (faces).
 - ✓ chaque ligne de ce bloc est caractérisée par les différents sommets constituant la facette en question.

Vision 3D

Représentation surfacique des objets 3D

	x	y	z
vertices:sommets	v 0.184420	0.850006	-0.488804
2	v 0.163326	0.747582	-0.640755
4	v 0.211223	0.770279	-0.594166
5	v 0.424938	0.253857	-0.158346
	f 1 2 3		
	f 4 3 2		
	f 2 5 4		

faces:facettes

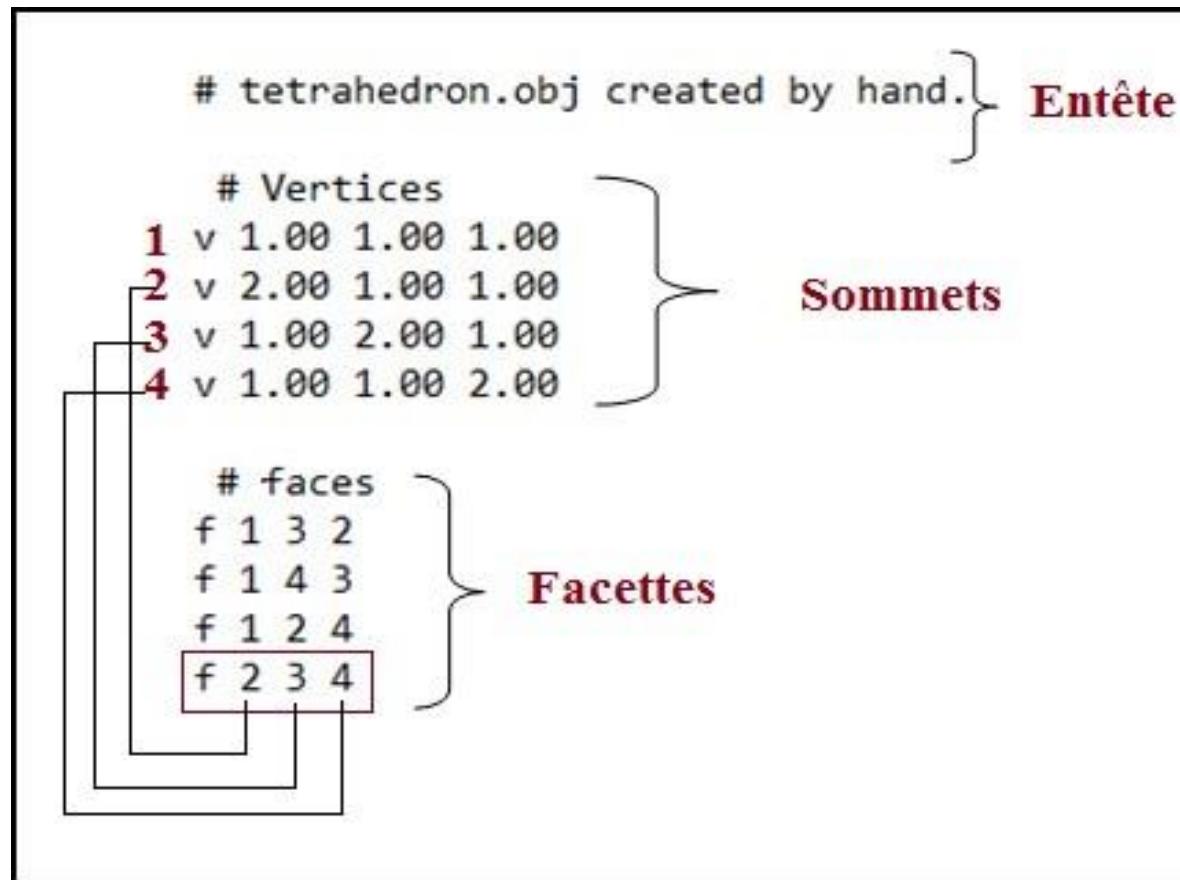
- ❖ Les coordonnées du premier sont (0. 184420, 0. 850006, -0. 488804)
- ❖ La dernière facette est composée du deuxième, cinquième et quatrième point

Extrait d'un fichier .obj

Vision 3D

Représentation surfacique des objets 3D

```
# tetrahedron.obj created by hand.  
# Vertices  
1 v 1.00 1.00 1.00  
2 v 2.00 1.00 1.00  
3 v 1.00 2.00 1.00  
4 v 1.00 1.00 2.00  
  
# faces  
f 1 3 2  
f 1 4 3  
f 1 2 4  
f 2 3 4
```



Entête

Sommets

Facettes

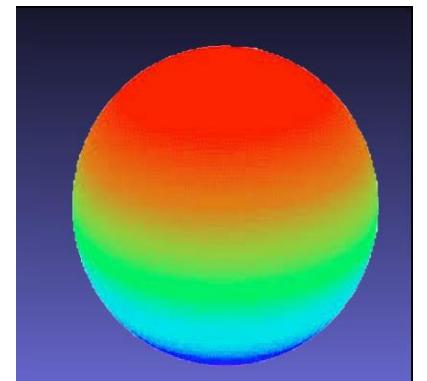
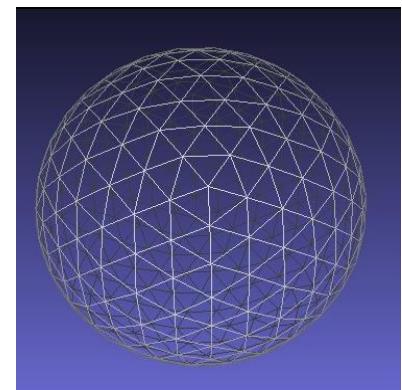
Extrait d'un fichier .obj

Vision 3D

Représentation surfacique des objets 3D

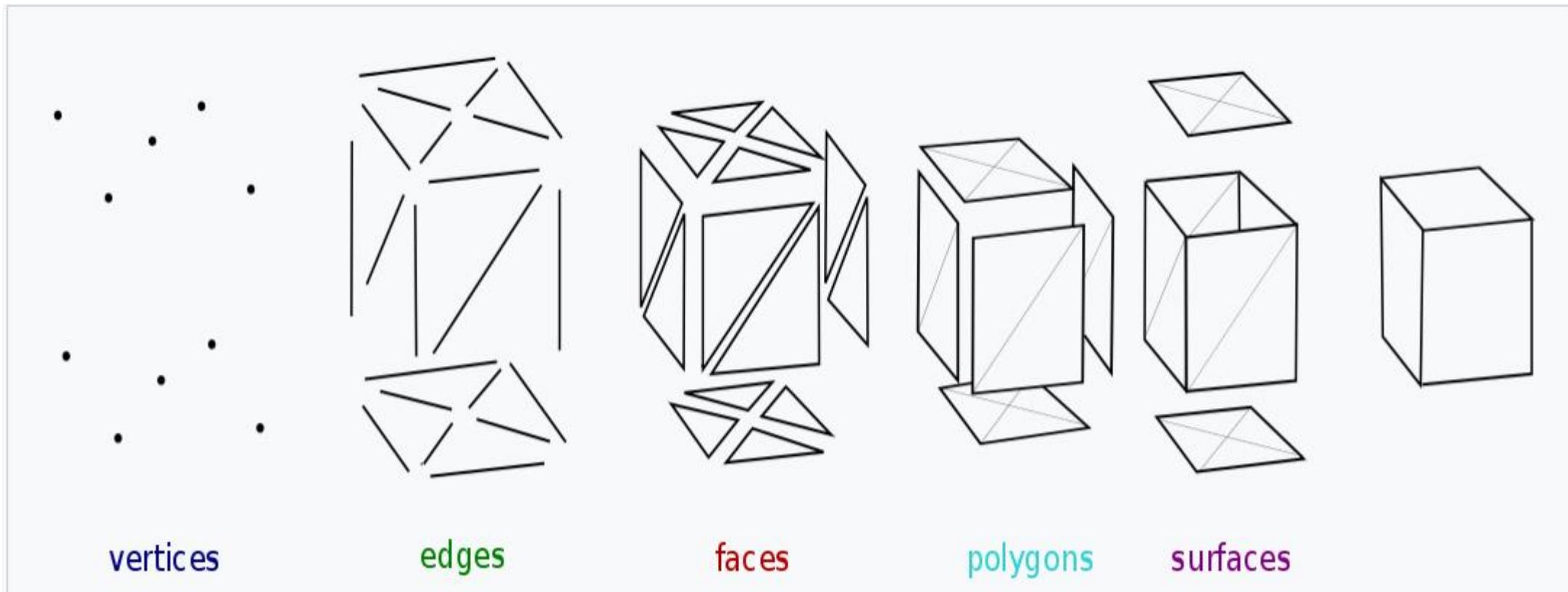
IV. Rendu 3D

- Pour visualiser un objet 3D, il existe plusieurs modes de rendu qui sont : rendu par nuage de points, rendu fil de fer et rendu surfacique.
- Avec le rendu nuage de points, seul l'ensemble des sommets du maillage est affiché
- Pour le rendu fil de fer, l'ensemble des polygones (dans notre cas les triangles) sont affichés, plus précisément sont affichés les sommets et les arêtes
- Le rendu surfacique consiste à attribuer une couleur à chaque point de la surface de l'objet, selon un modèle d'ombrage précis comme par exemple l'ombrage constant, l'ombrage de Gouraud et la celui de Phong.



Vision 3D

Représentation surfacique des objets 3D



A la prochaine !