

Point Processing & Modeling

Sources



LIDAR

Batiments, etc

Sources



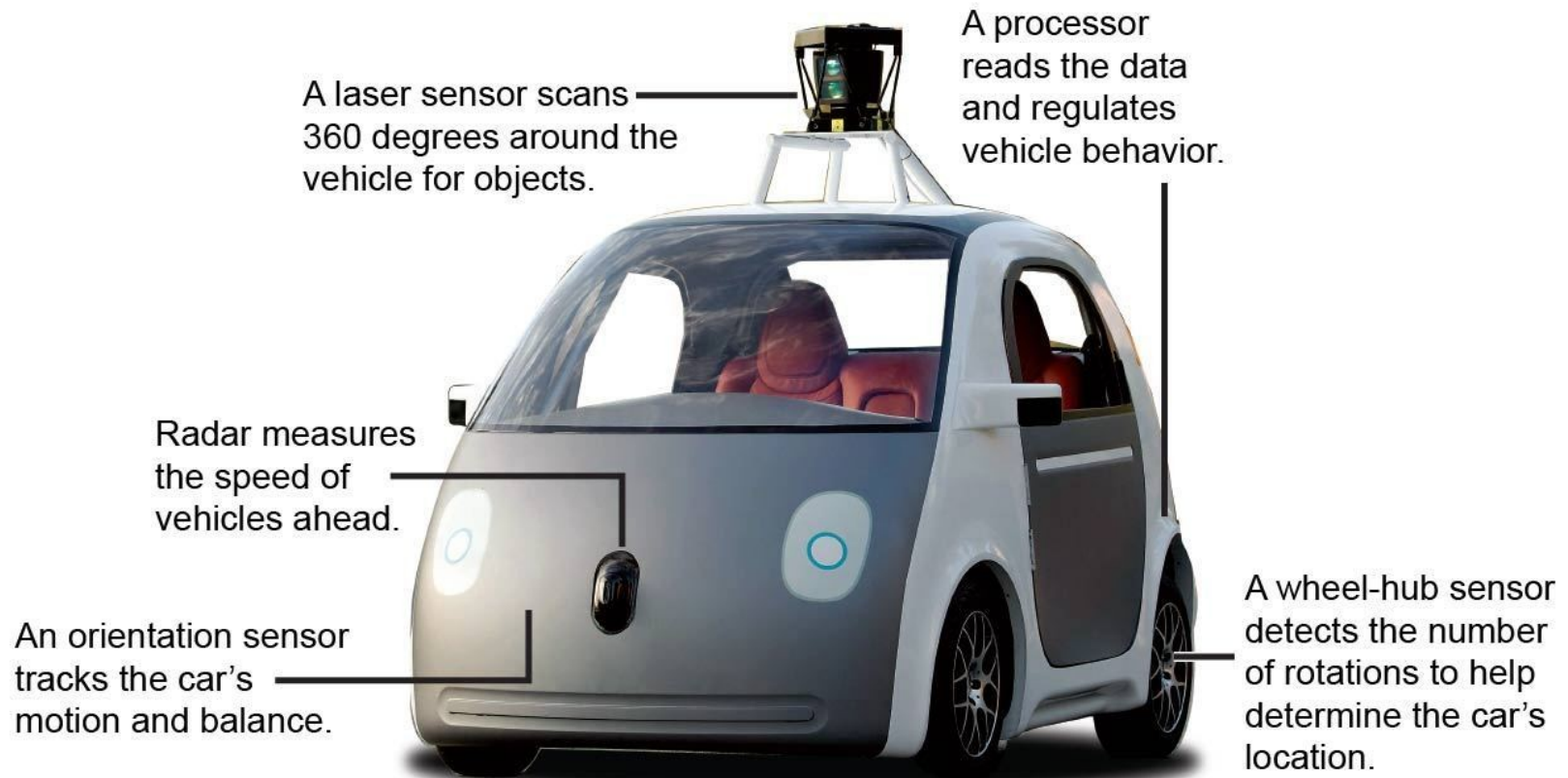
Table tournantes

Sources



Kinect

Sources

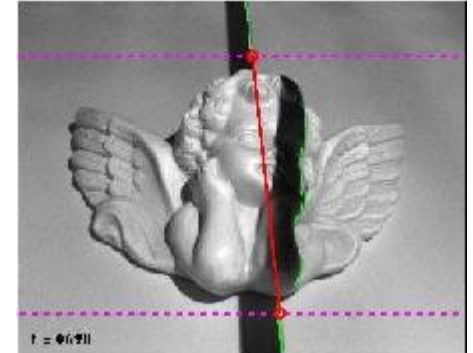
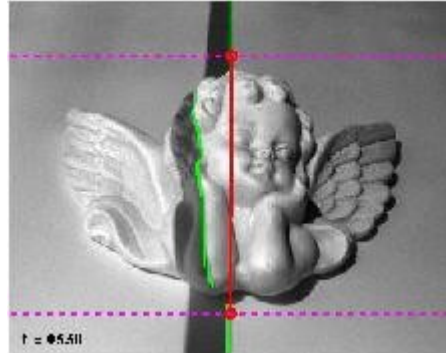


Source: Google

Raoul Rañoa / @latimesgraphics

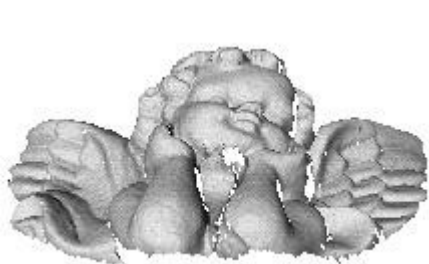
Google car

Sources

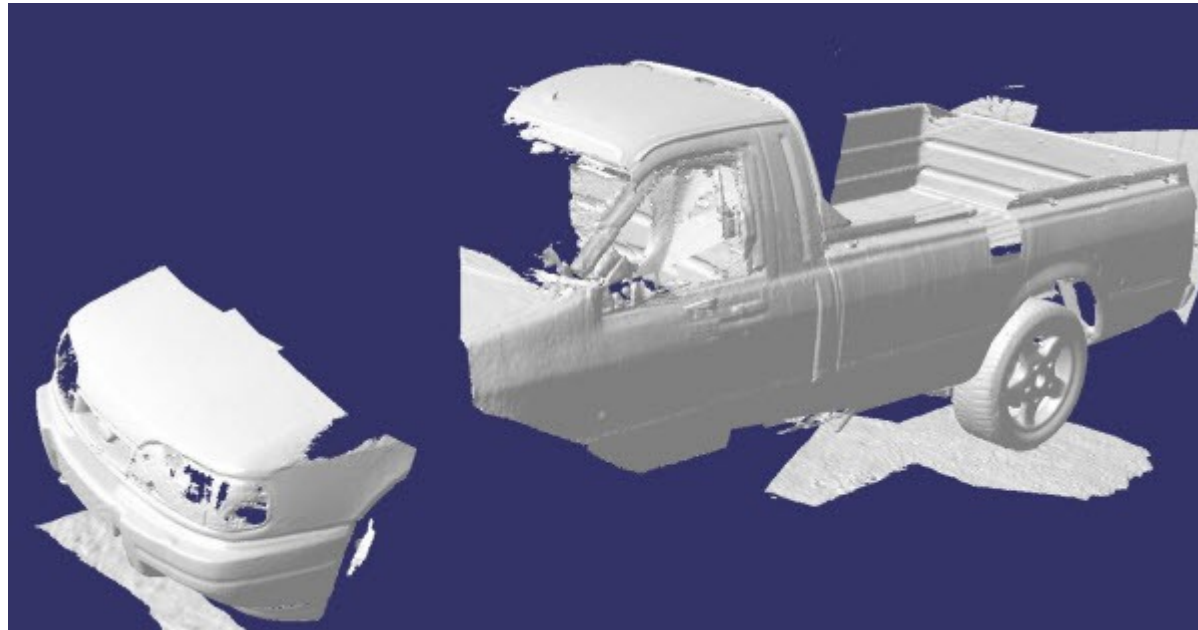


Système fait avec une lampe et un crayon...

<http://www.vision.caltech.edu/bouguetj/ICCV98/>



Bruits et propriétés géométriques



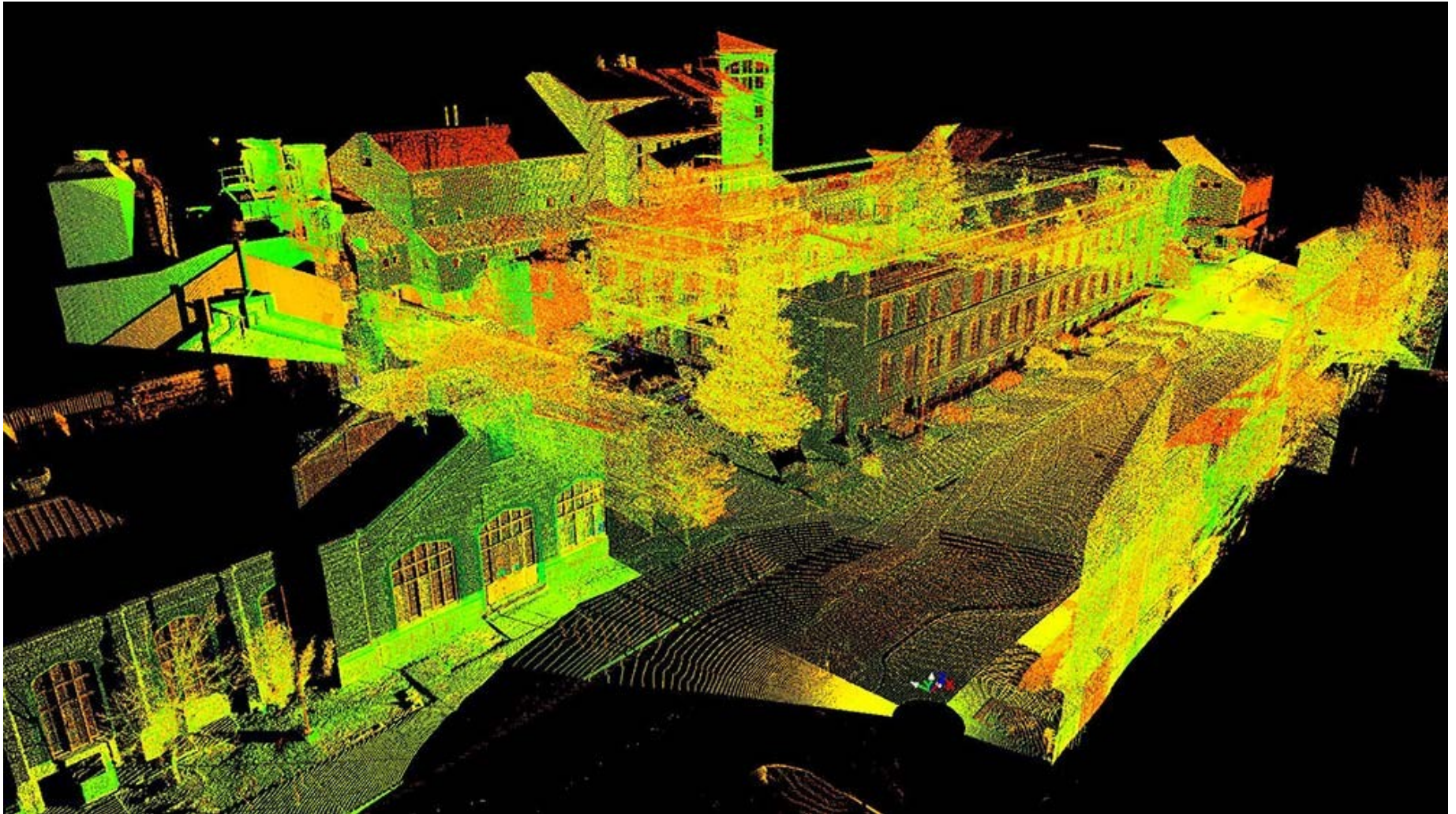
Occlusions, parties transparentes ou réfléchissantes

Bruits et propriétés géométriques



Quantification, popping

Bruits et propriétés géométriques



Direction du scan
<http://www.adamdealva.com/>

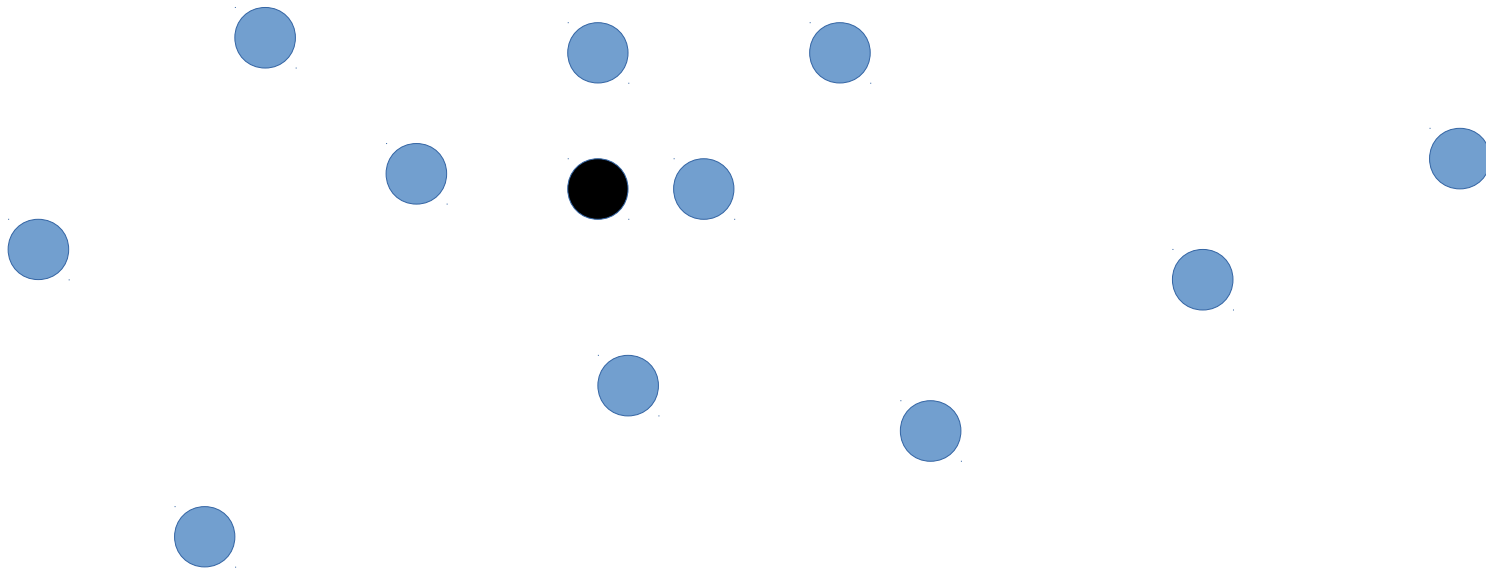
Bruits et propriétés géométriques

Différents types de bruit, différents artifacts

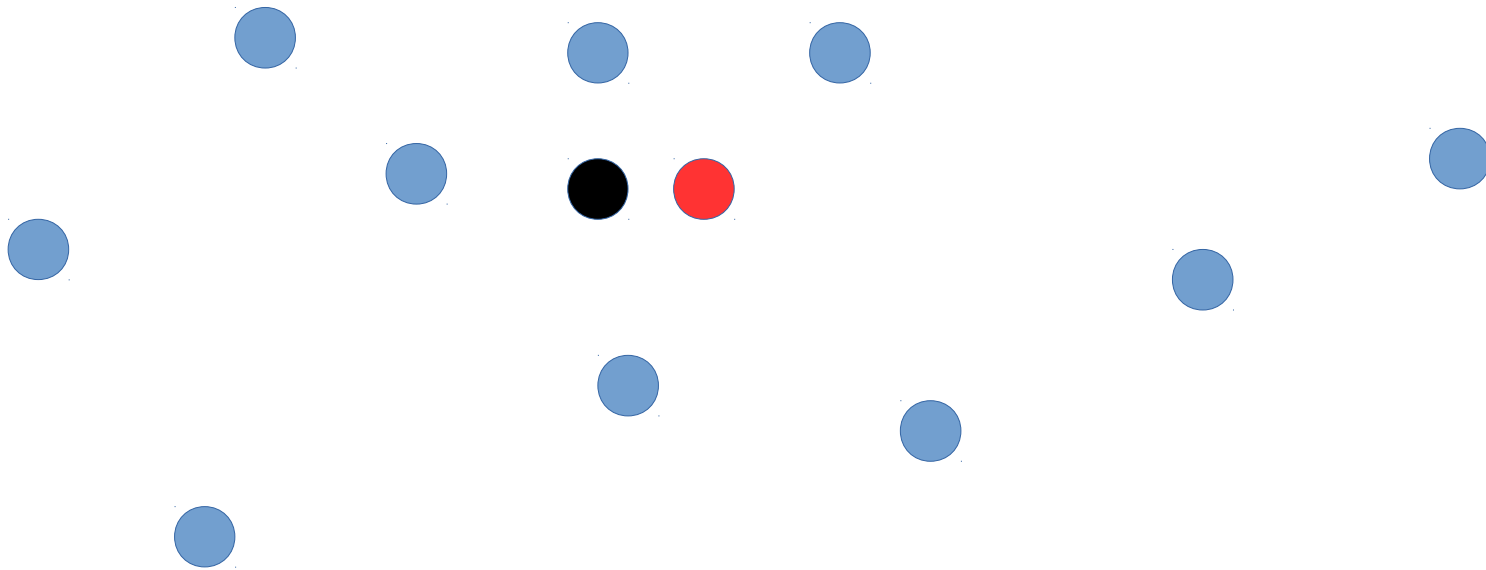
→

Différents algorithmes de traitement

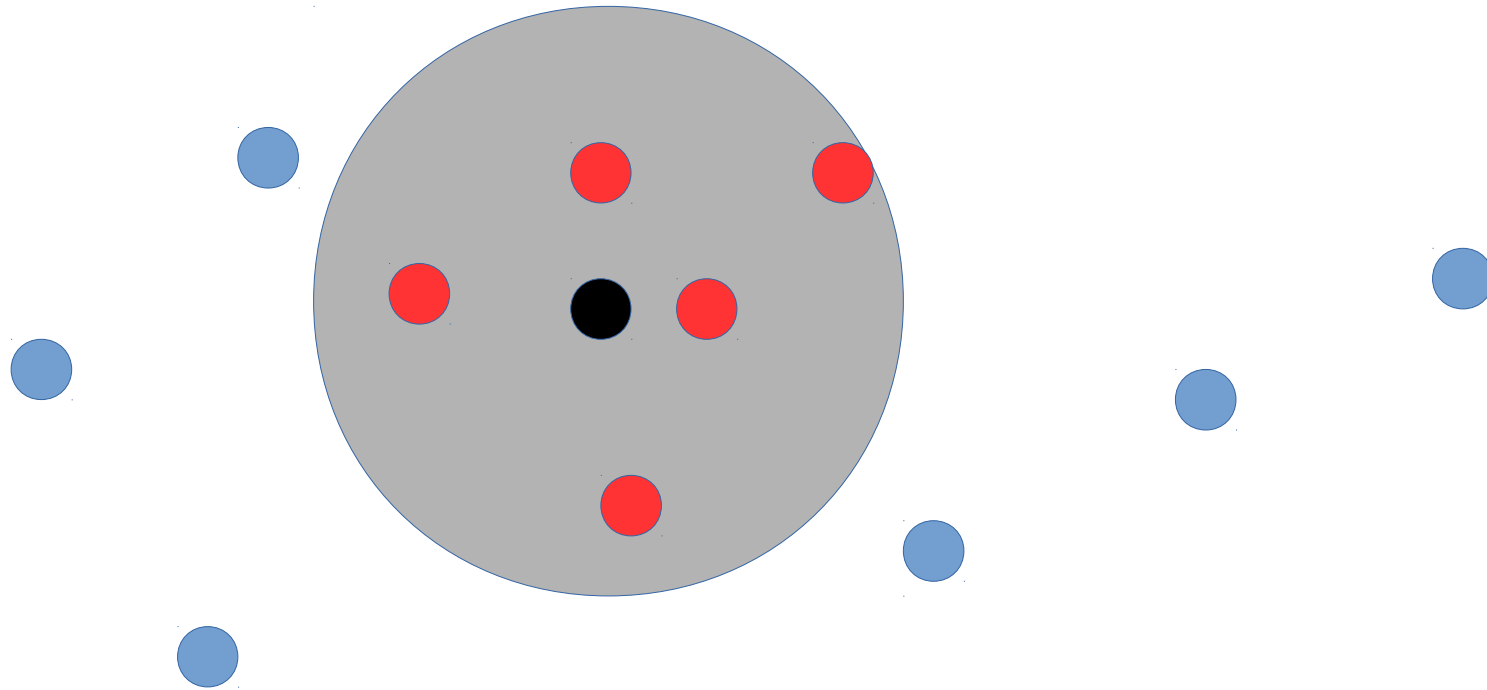
Recherche de voisinages Euclidiens



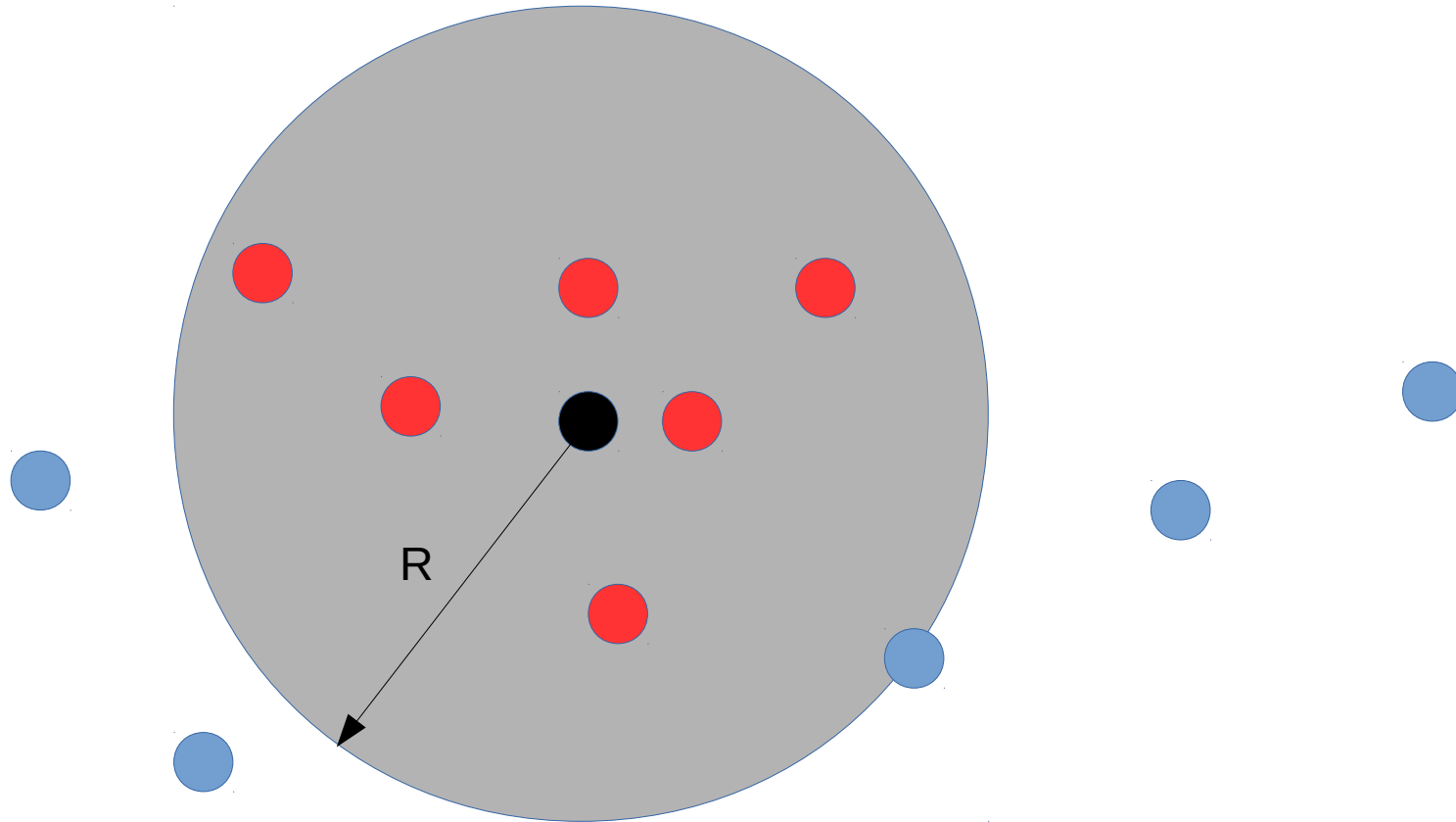
Plus proche voisin



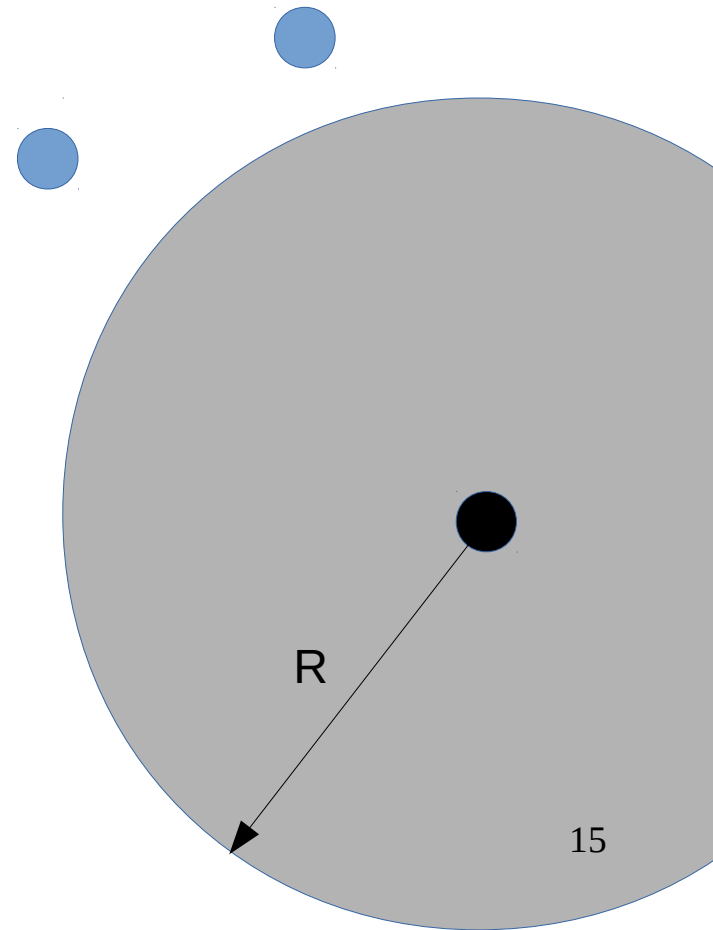
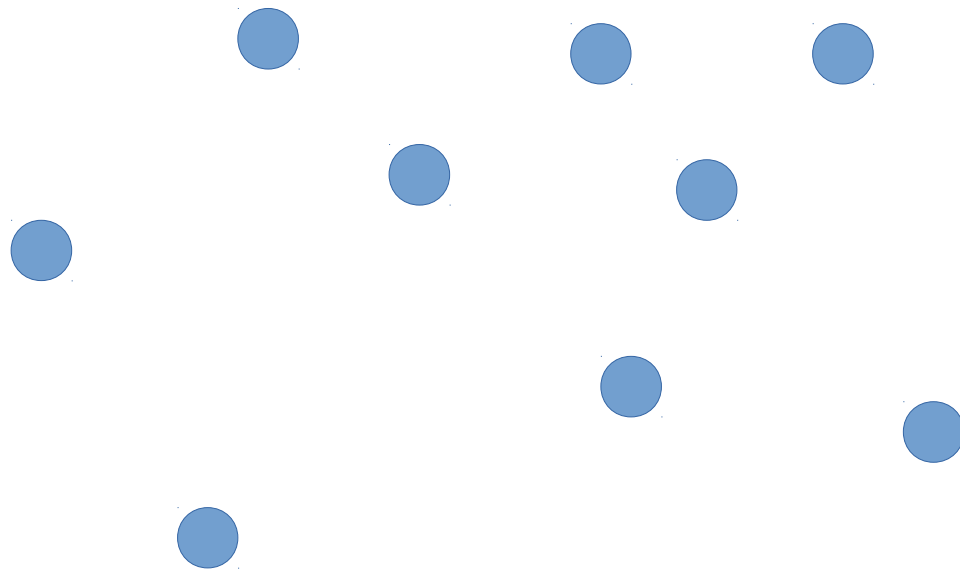
K (ici, 5) plus proches voisins



Voisins a distance $< R$

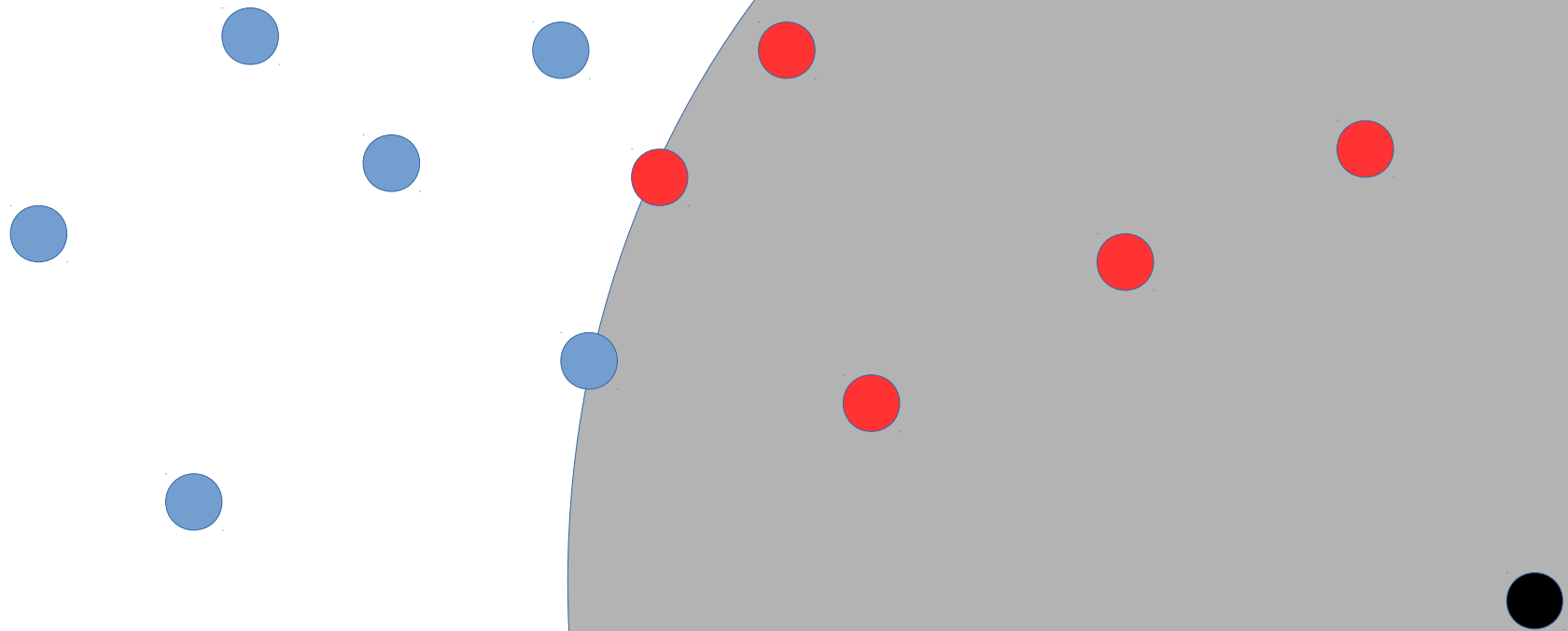


Voisins a distance $< R$



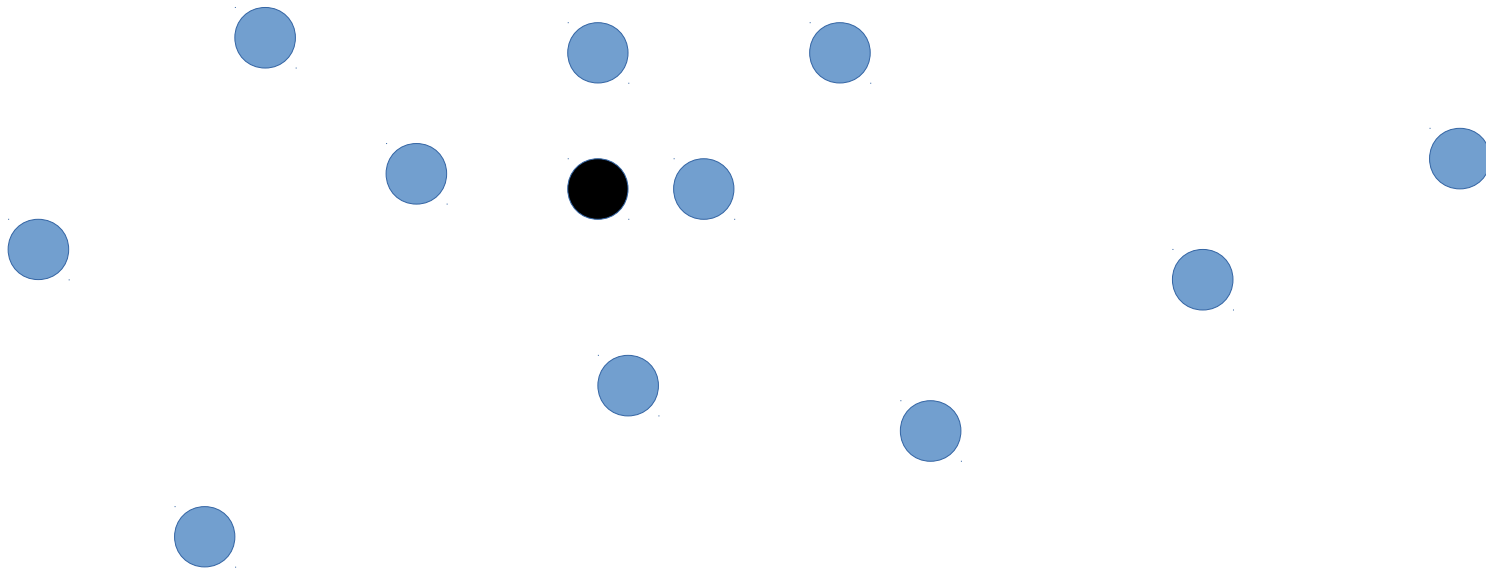
Voisinage dans $B(p, R)$ peut être vide

K (ici, 5) plus proches voisins



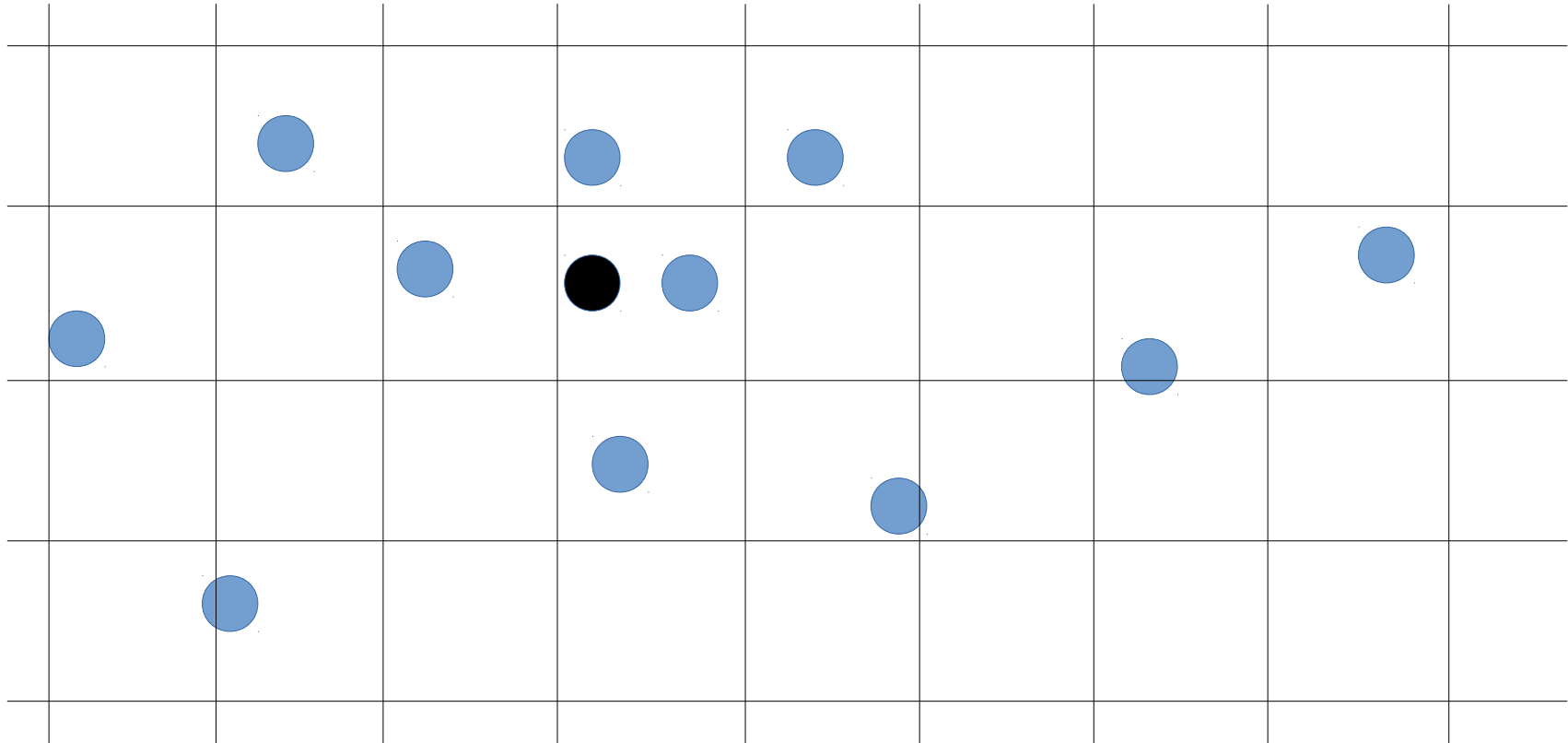
K plus proches voisins peuvent être dans un voisinage très éloigné, et mal distribués !

Recherche de voisinages Euclidiens : Structures



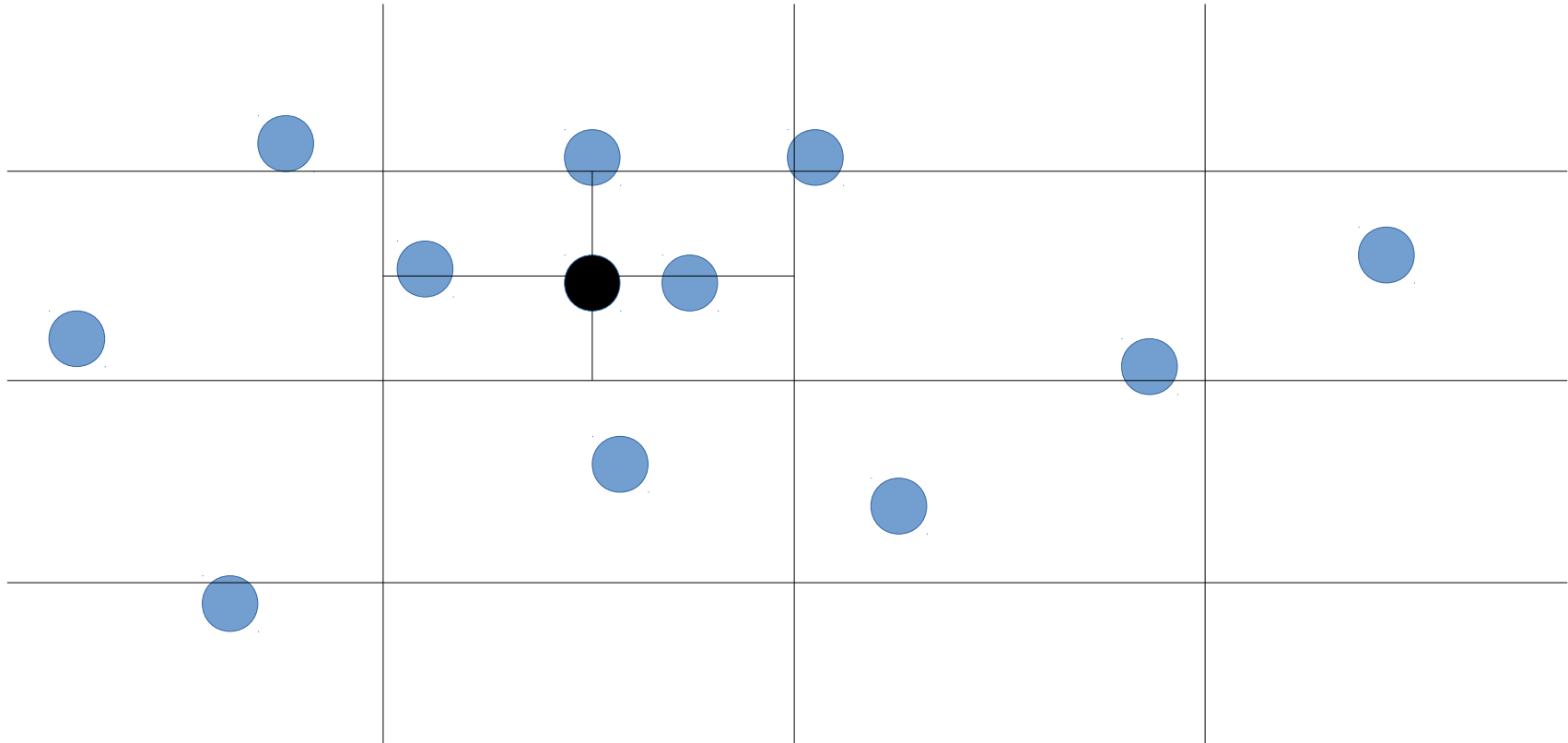
Parcours linéaire : le plus simple, le plus stupide,
aucune structure à stocker.

Recherche de voisinages Euclidiens : Structures



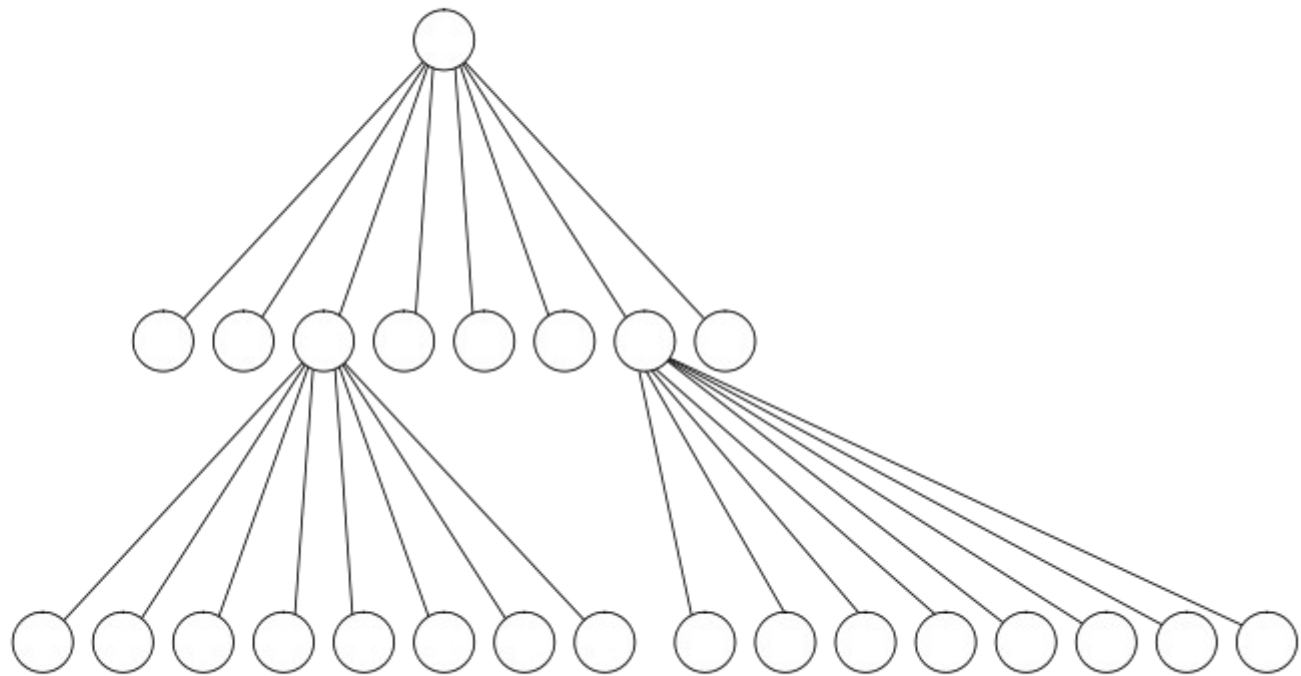
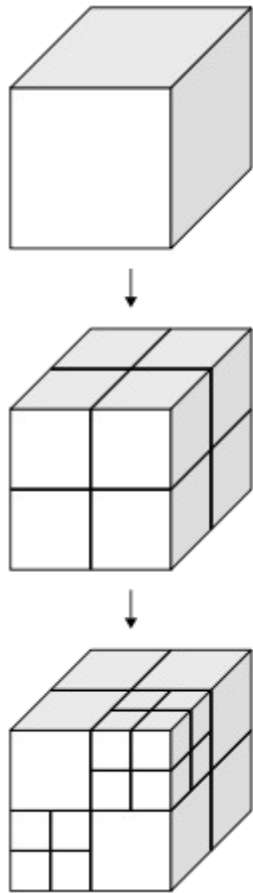
Grille uniforme : très simple, dépend de la taille de la cellule, gâche des espaces vides → pas adaptés à des points distribués sur une surface

Recherche de voisinages Euclidiens : Structures

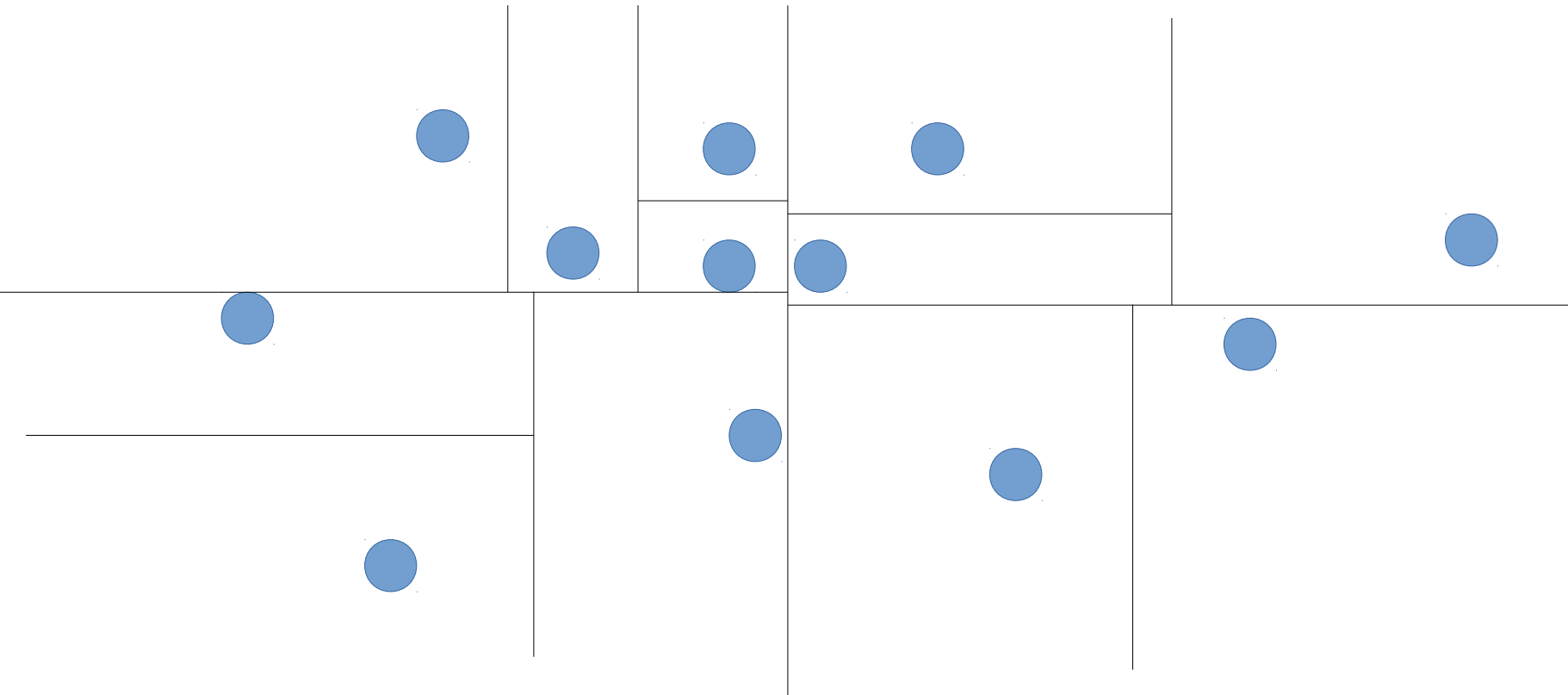


Grille hiérarchique (quadtree en 2D, octree en 3D):
construction simple, query simple, adaptatif → adapté a
des points distribués sur une surface

Représentation « sparse » des octrees

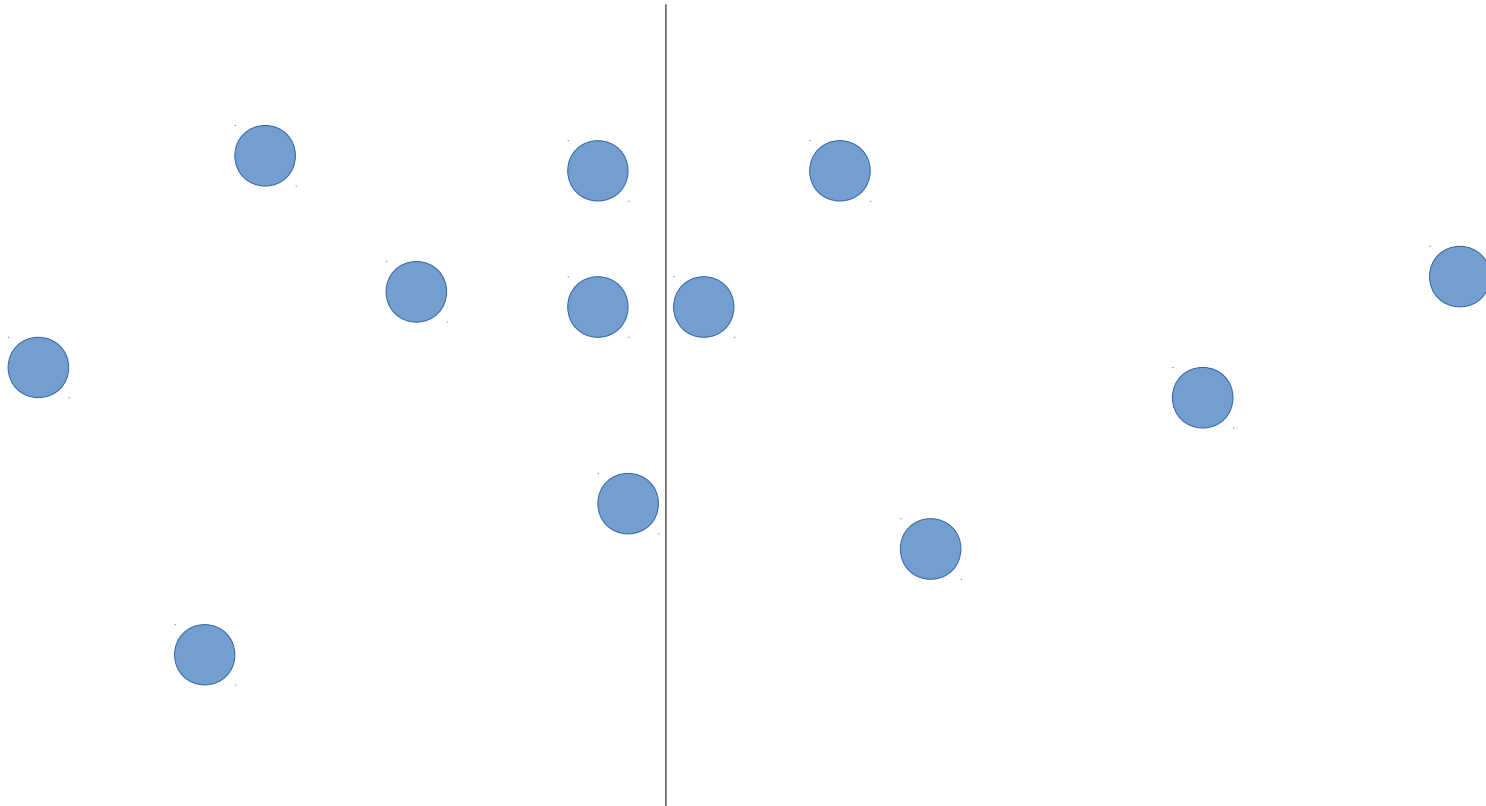


kd-tree



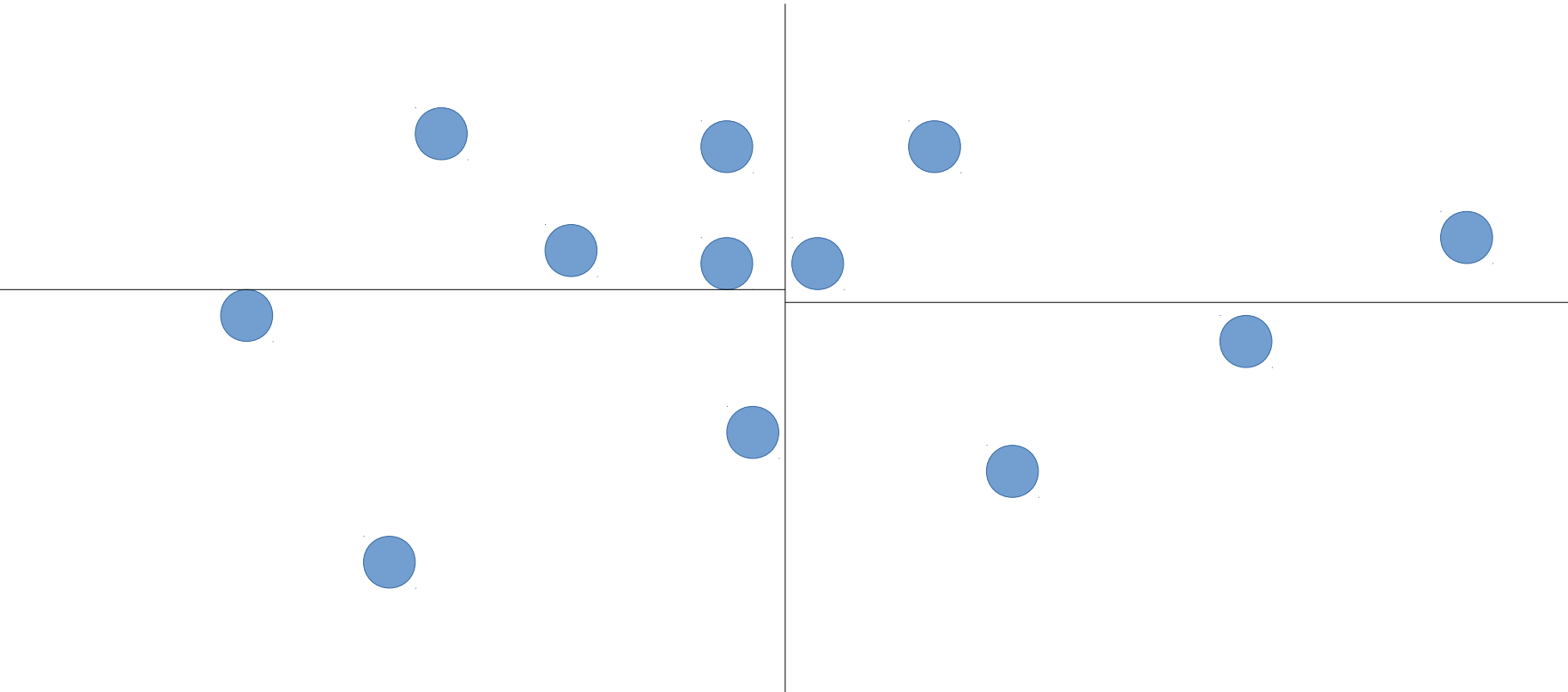
kd-tree: construction simple, query un peu compliquée
mais efficace, adaptatif → adapté a des points
distribués sur une surface

kd-tree



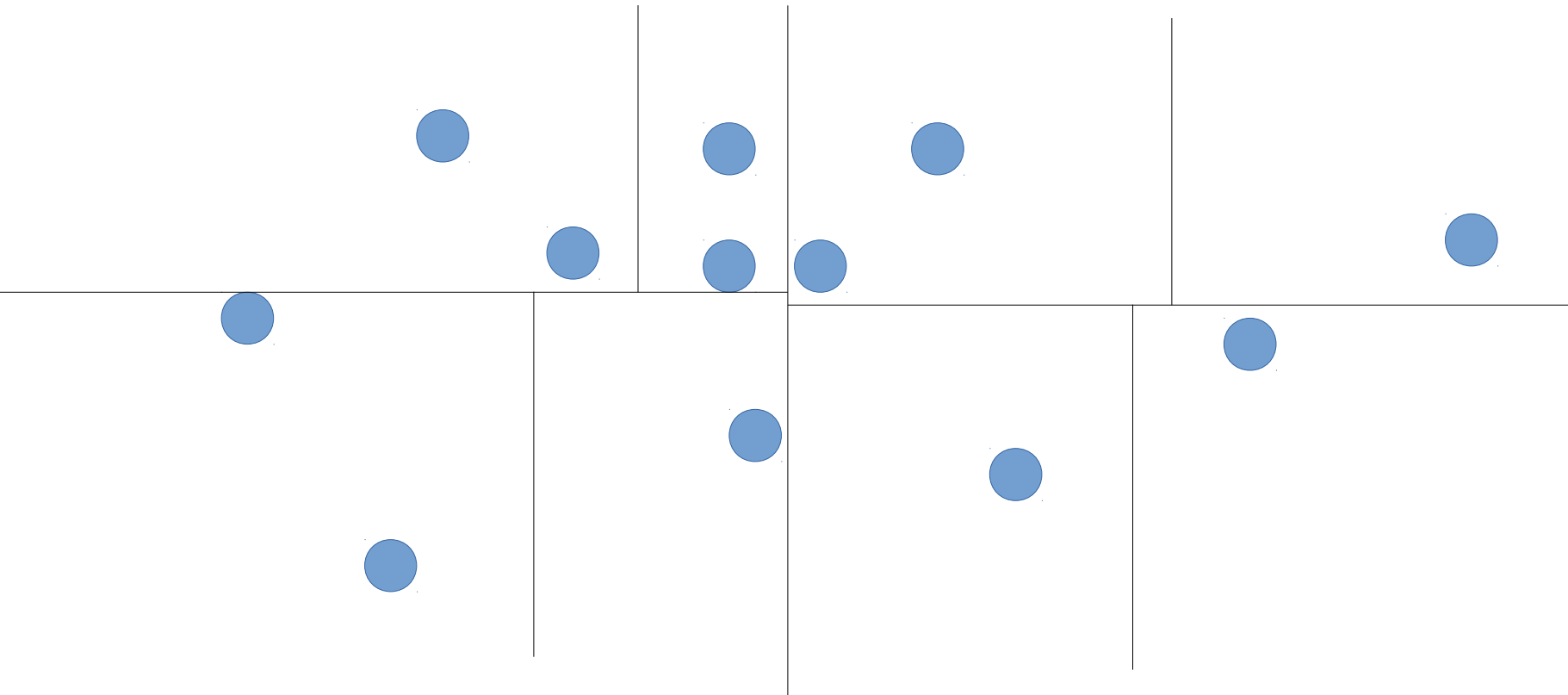
kd-tree: construction simple

kd-tree



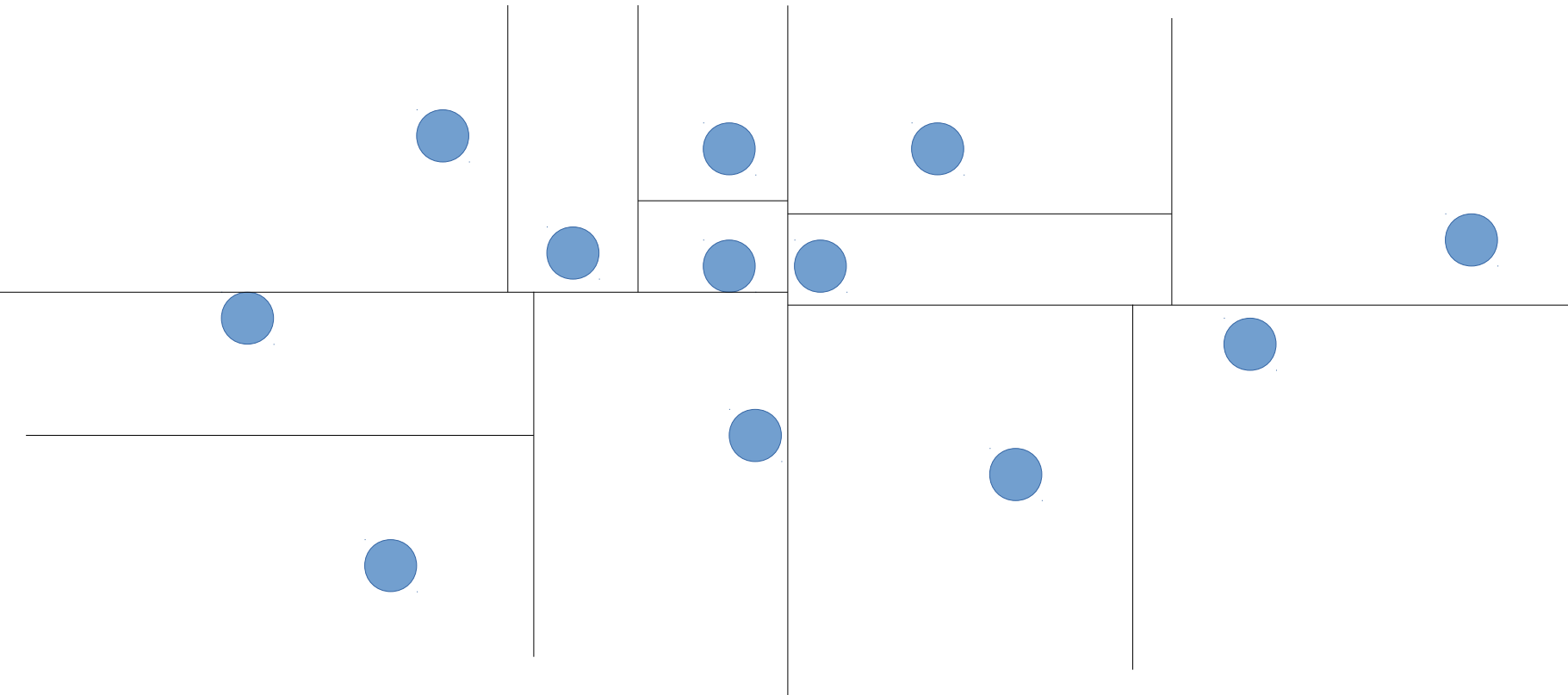
kd-tree: construction simple

kd-tree



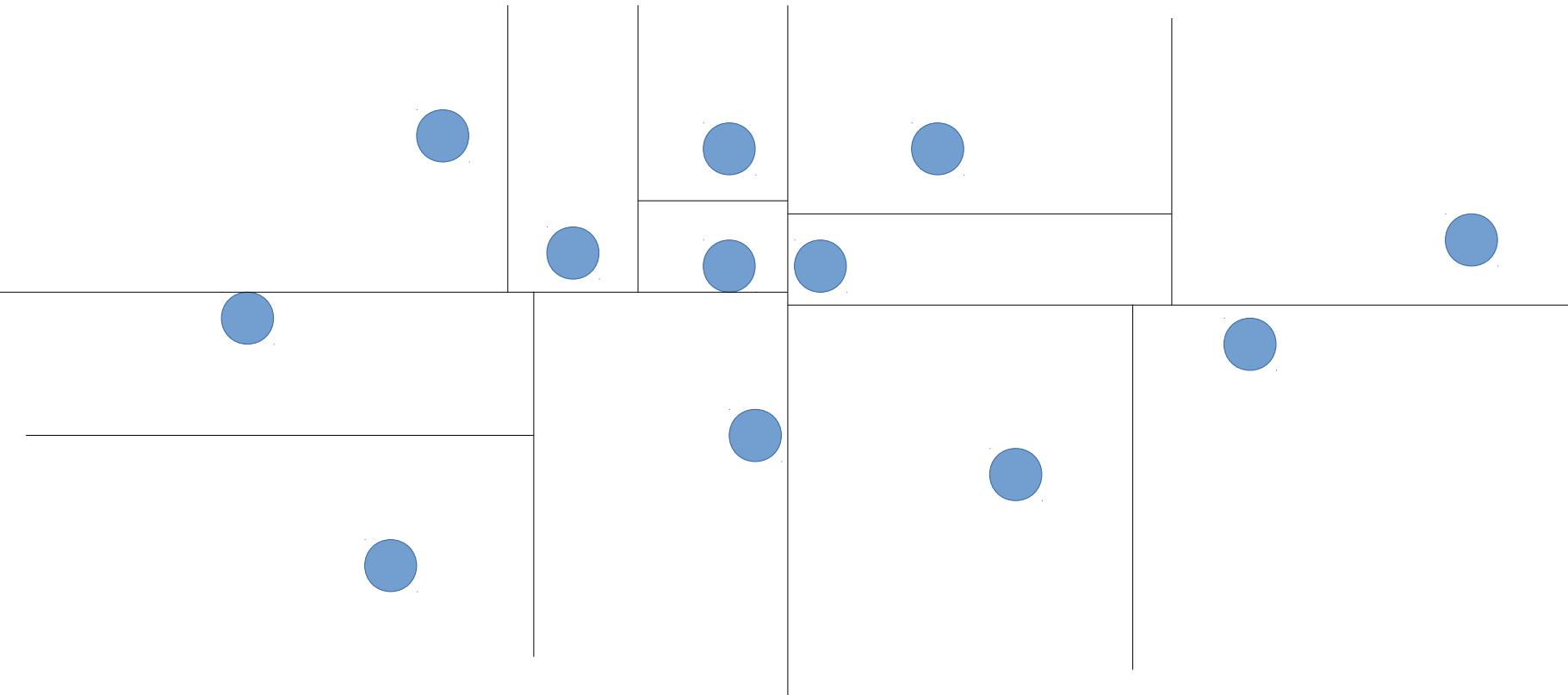
kd-tree: construction simple

kd-tree



kd-tree: construction simple

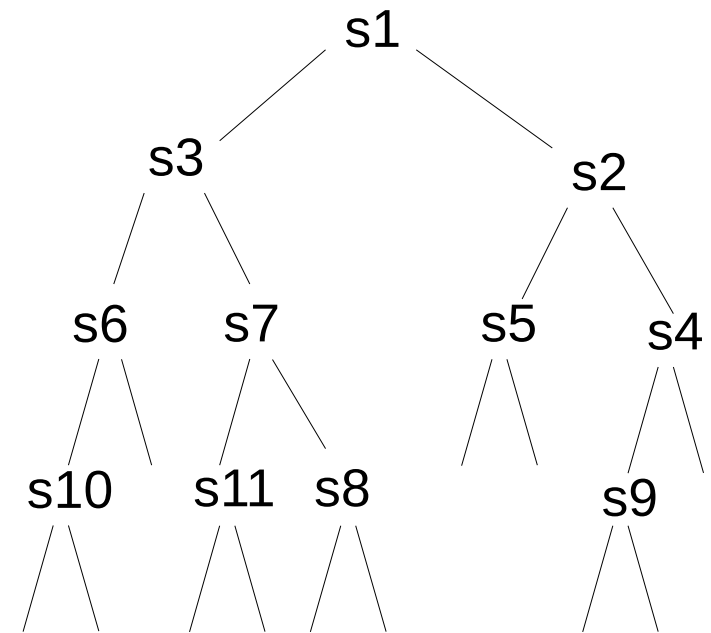
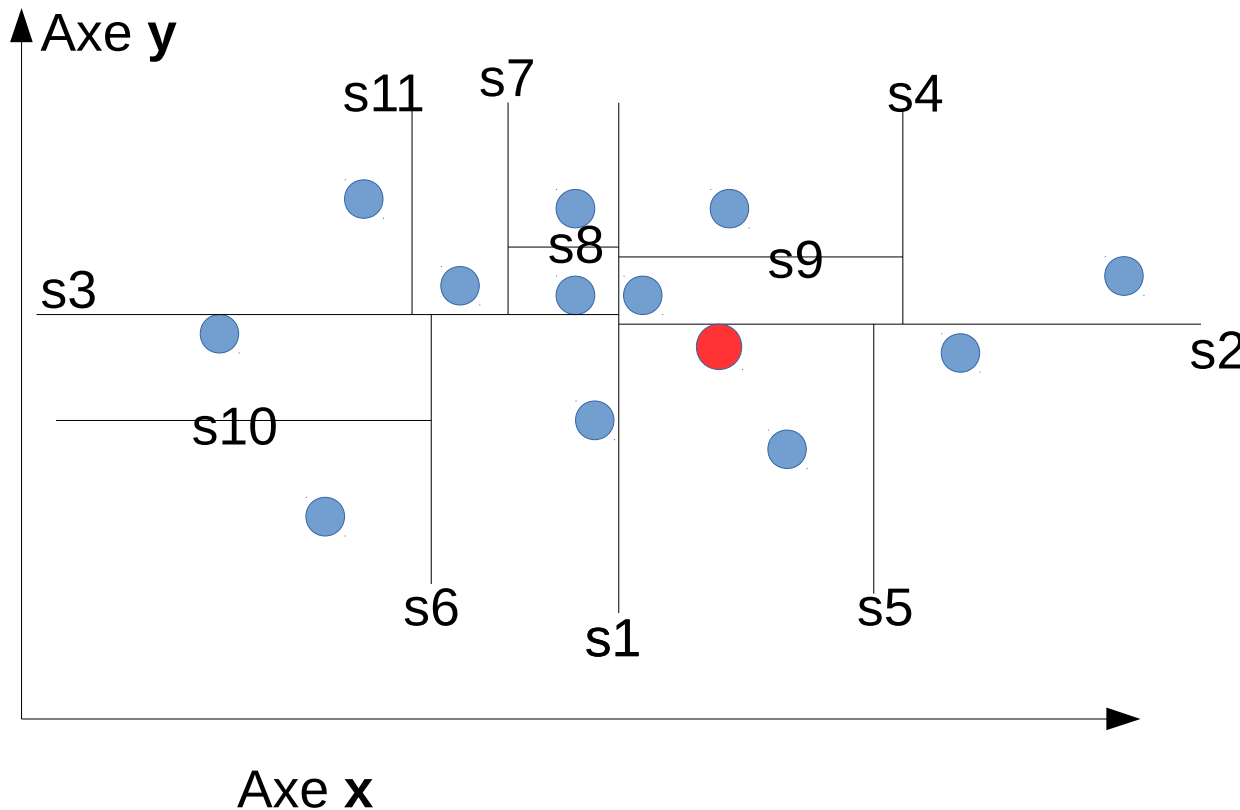
kd-tree



kd-tree: chaque nœud stocke l'axe de subdivision (par ex, « x »), une équation décrivant sa géométrie (par ex, « $x=4$ »), et des pointeurs vers ses deux nœuds fils

Kd-tree : recherche du plus proche voisin

- Identifier la feuille dans laquelle tombe le point
- En remontant l'arbre, vérifier si les cellules voisines peuvent contenir des points plus proches



Kd-tree : recherche du plus proche voisin

```
NNS(q: point, n: node, p: point, w: distance) : point {
  if n.left = null then {leaf case}
    if distance(q,n.point) < w then return n.point else return p;
  else
    if w = infinity then
      if q(n.axis) ≤ n.value then
        p := NNS(q,n.left,p,w);
        w := distance(p,q);
        if q(n.axis) + w > n.value then p := NNS(q, n.right, p, w);
      else
        p := NNS(q,n.right,p,w);
        w := distance(p,q);
        if q(n.axis) - w ≤ n.value then p := NNS(q, n.left, p, w);
    else //w is finite//
      if q(n.axis) - w ≤ n.value then
        p := NNS(q, n.left, p, w);
        w := distance(p,q);
        if q(n.axis) + w > n.value then p := NNS(q, n.right, p, w);
    return p
}
```

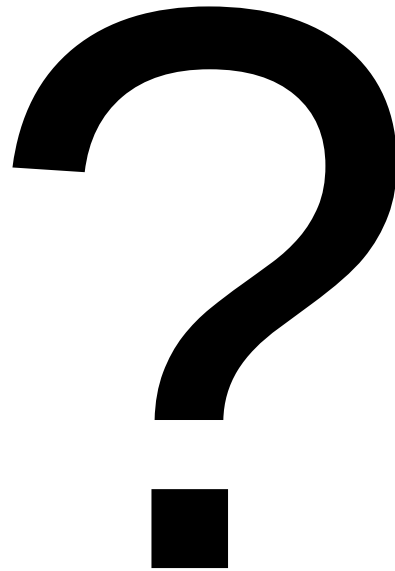
Ressources extérieures utilisées :

<https://courses.cs.washington.edu/courses/cse373/02au/lectures/lecture22l.pdf>

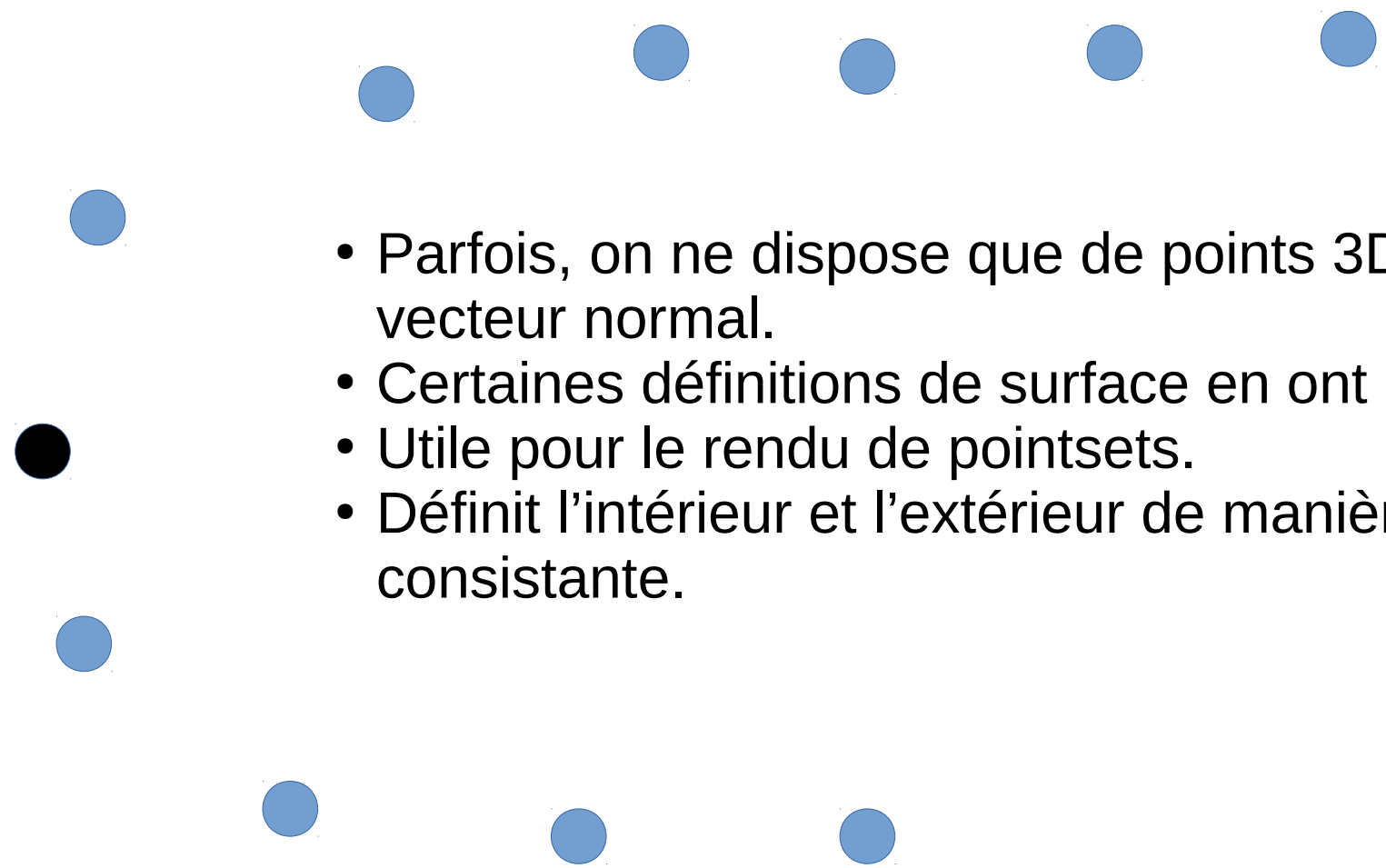
Kd-tree

- Permet la recherche du plus proche voisin
 - Permet la recherche des k plus proches voisins
 - Permet la recherche des voisins dans une boule
 - Complexite de construction : $n \log(n)$
 - Complexite de query : $\log(n)$
-
- Librairies open source existantes
 - Code fourni pour le TP de la semaine prochaine

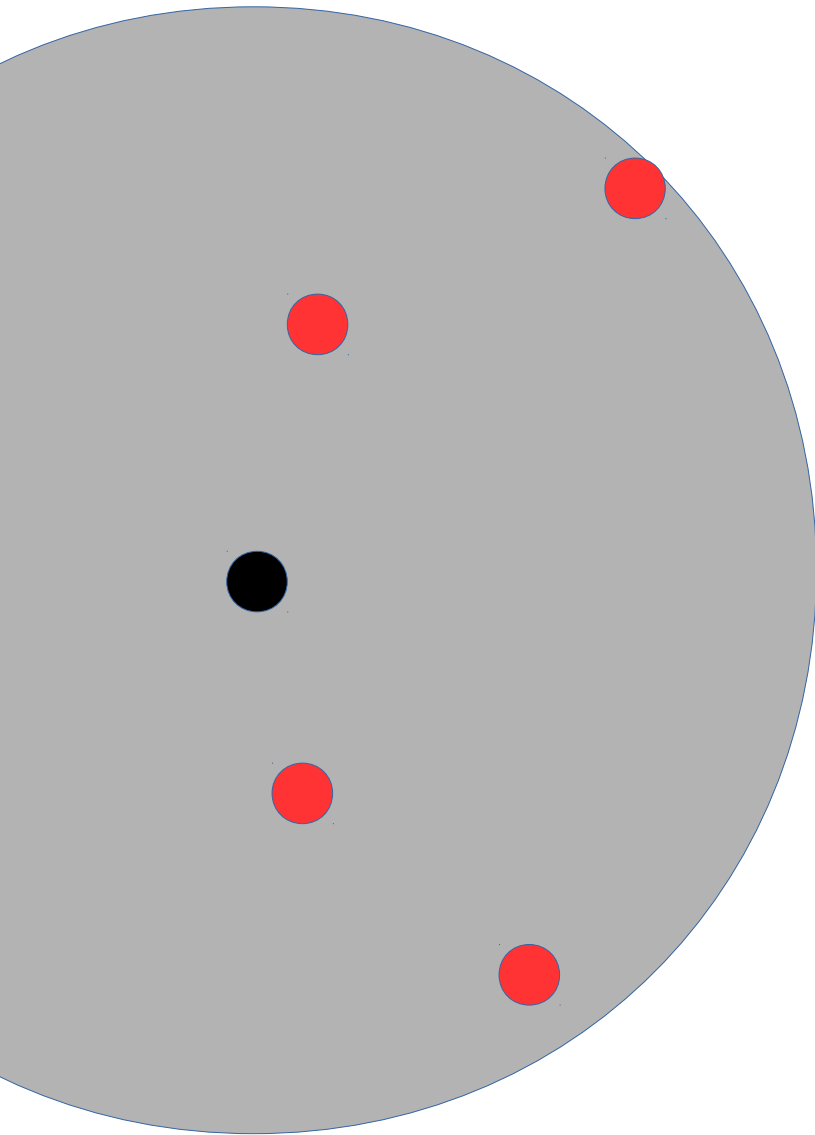
Questions ?



Estimation des normales

- 
- Parfois, on ne dispose que de points 3D, sans leur vecteur normal.
 - Certaines définitions de surface en ont besoin.
 - Utile pour le rendu de pointsets.
 - Définit l'intérieur et l'extérieur de manière consistante.

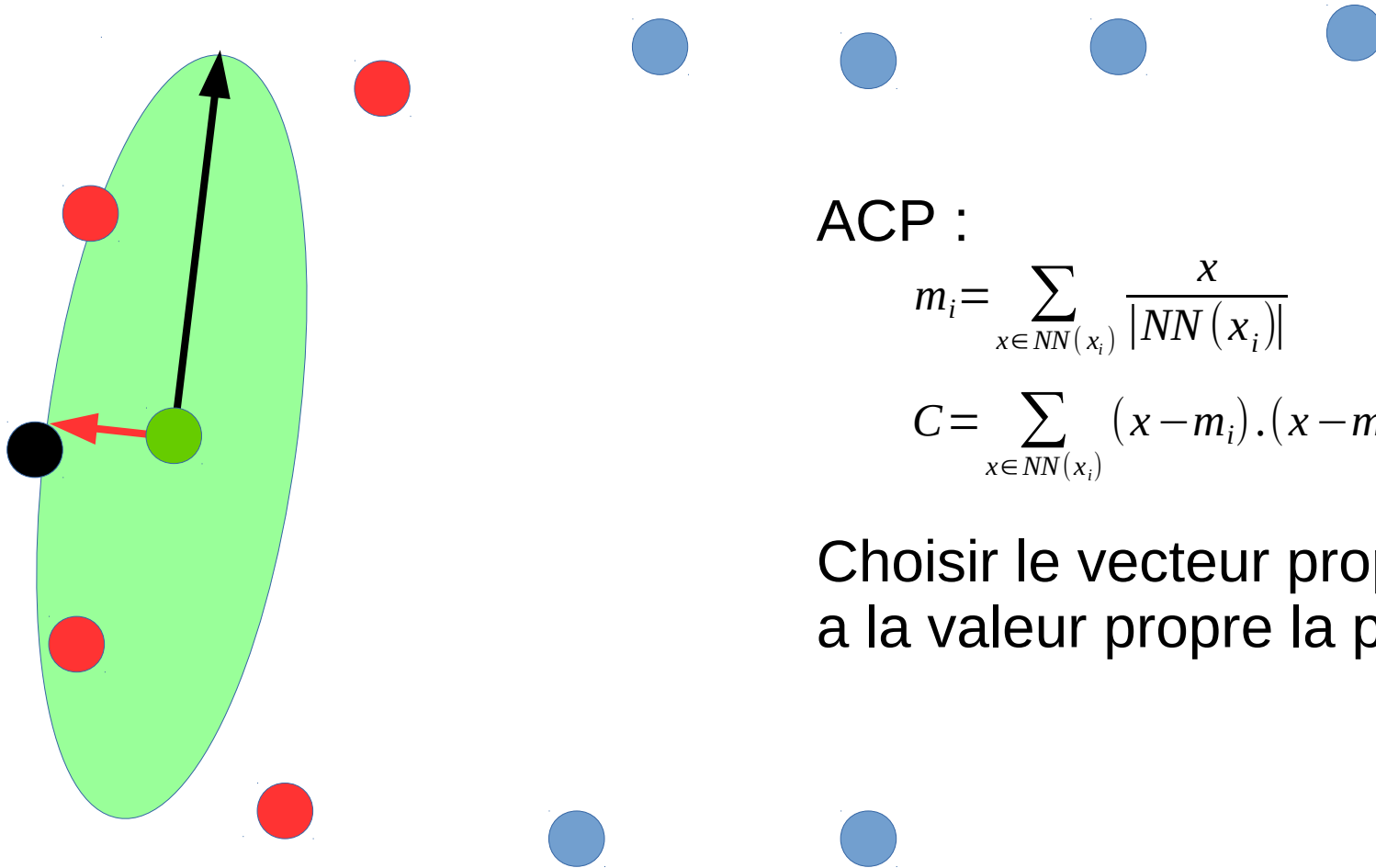
Estimation des normales



Plus proches voisins : $NN(x_i)$



Estimation des normales



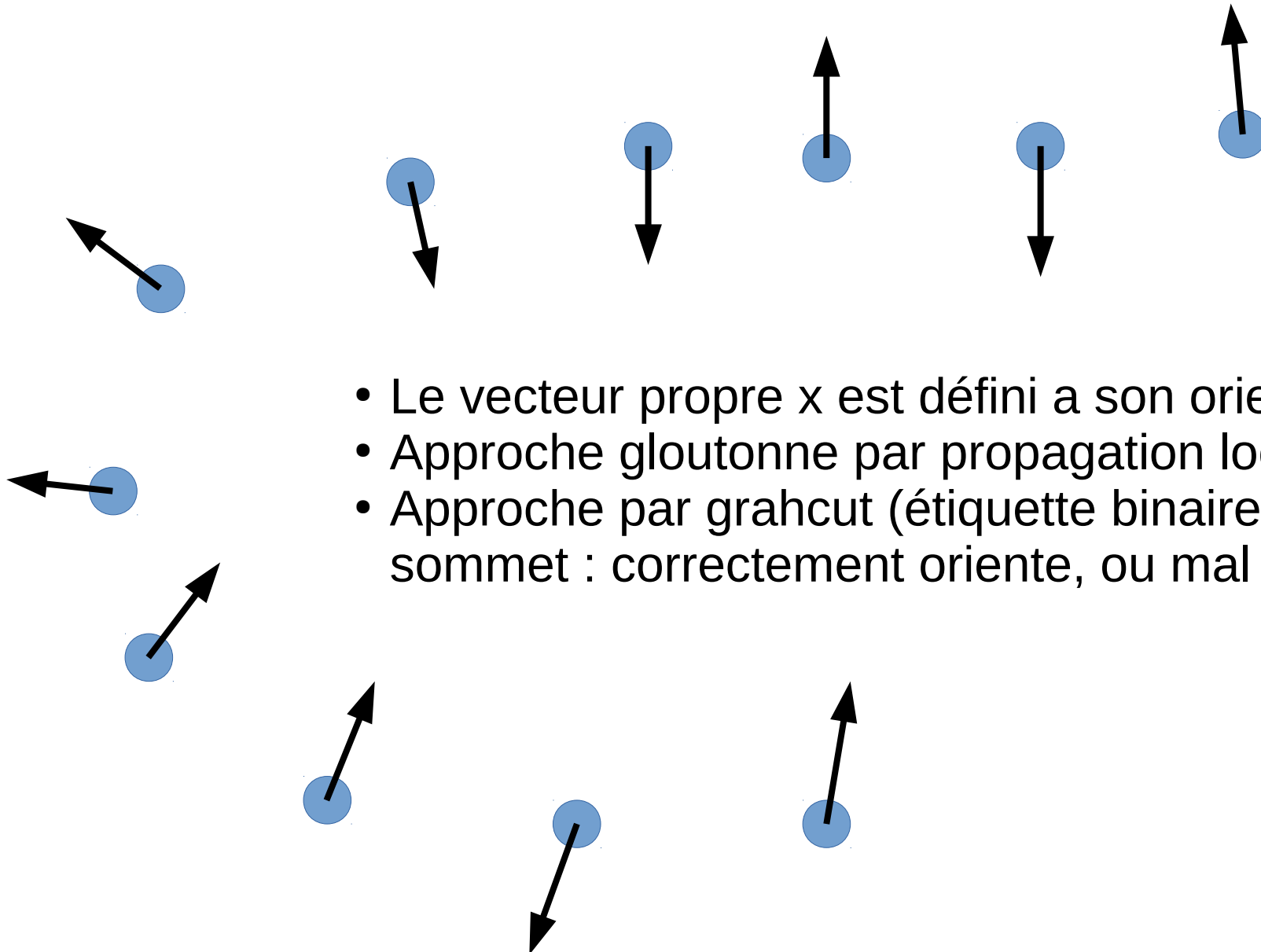
ACP :

$$m_i = \sum_{x \in NN(x_i)} \frac{x}{|NN(x_i)|}$$

$$C = \sum_{x \in NN(x_i)} (x - m_i) \cdot (x - m_i)^T$$

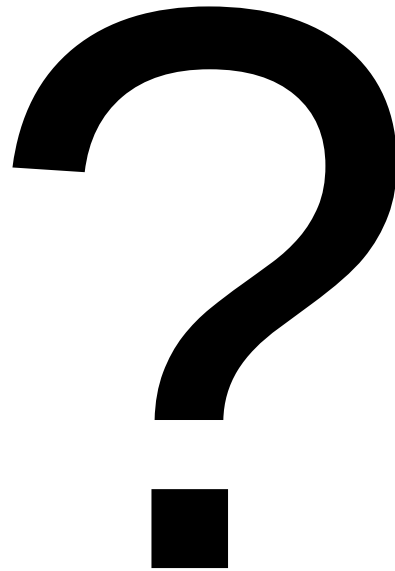
Choisir le vecteur propre associé à la valeur propre la plus petite.

Orientation consistente

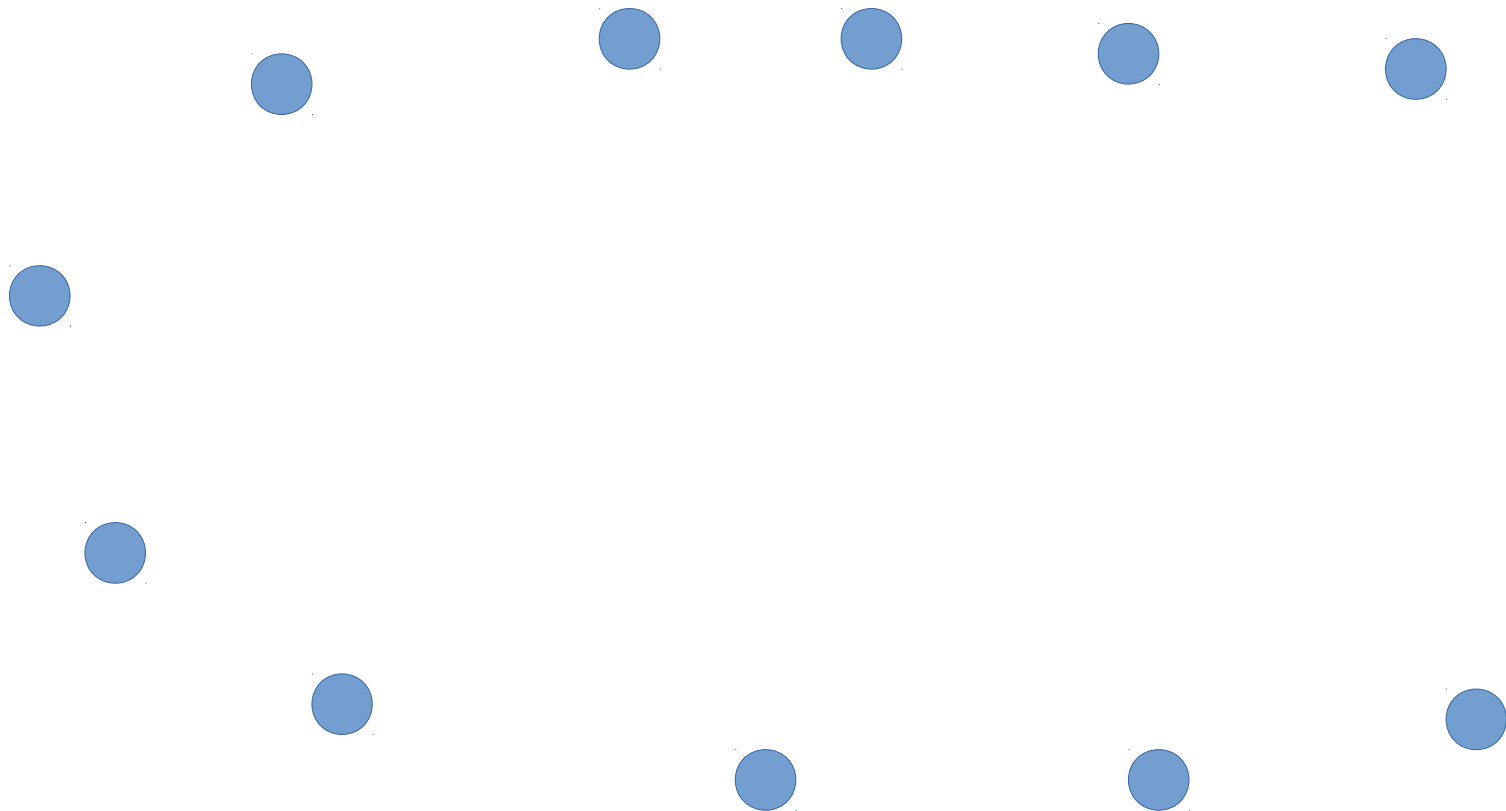


- Le vecteur propre x est défini a son orientation près
- Approche gloutonne par propagation locale
- Approche par grahcut (étiquette binaire pour chaque sommet : correctement oriente, ou mal oriente)

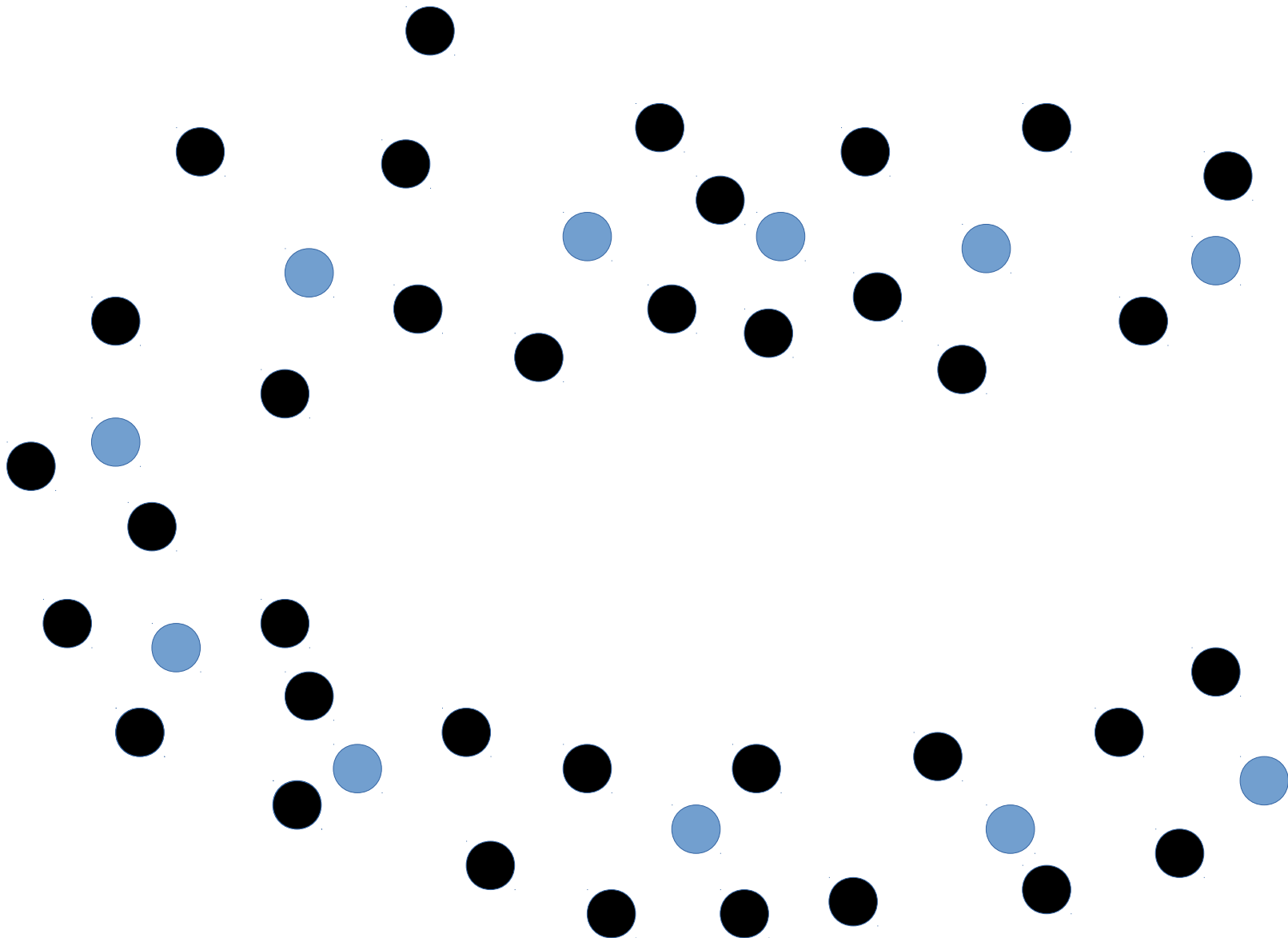
Questions ?



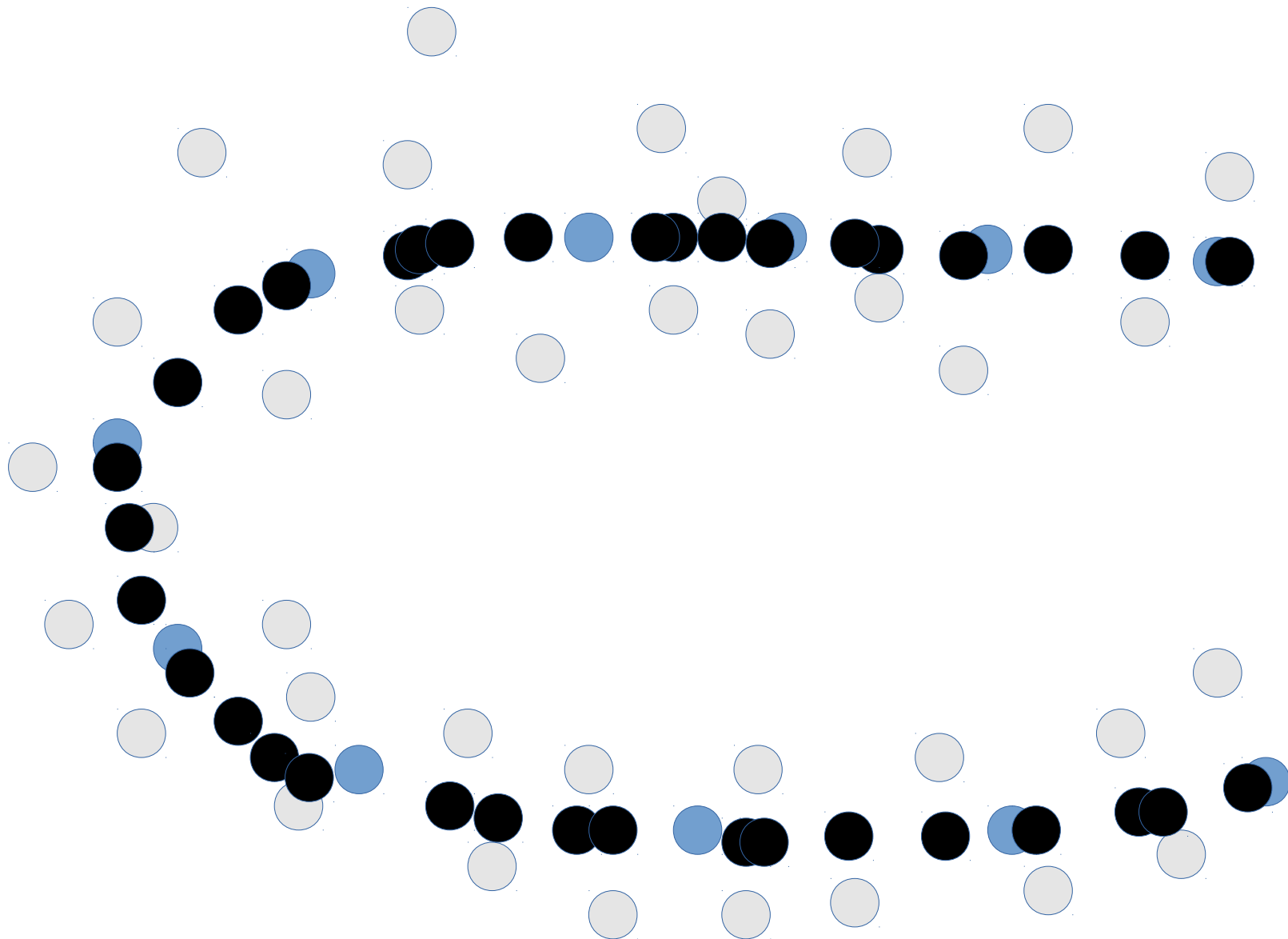
Modèles de surfaces de points



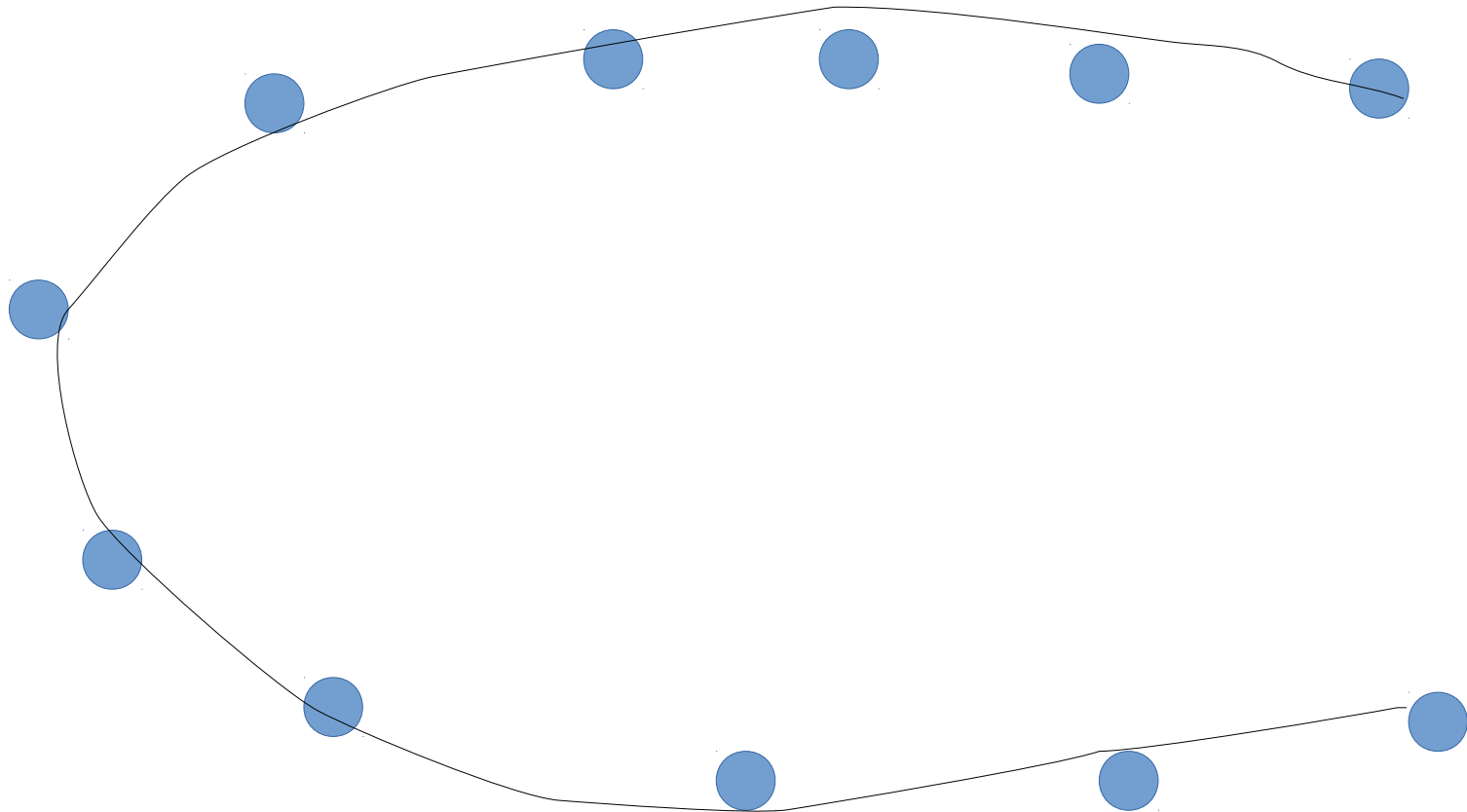
Définition par projection



Définition par projection

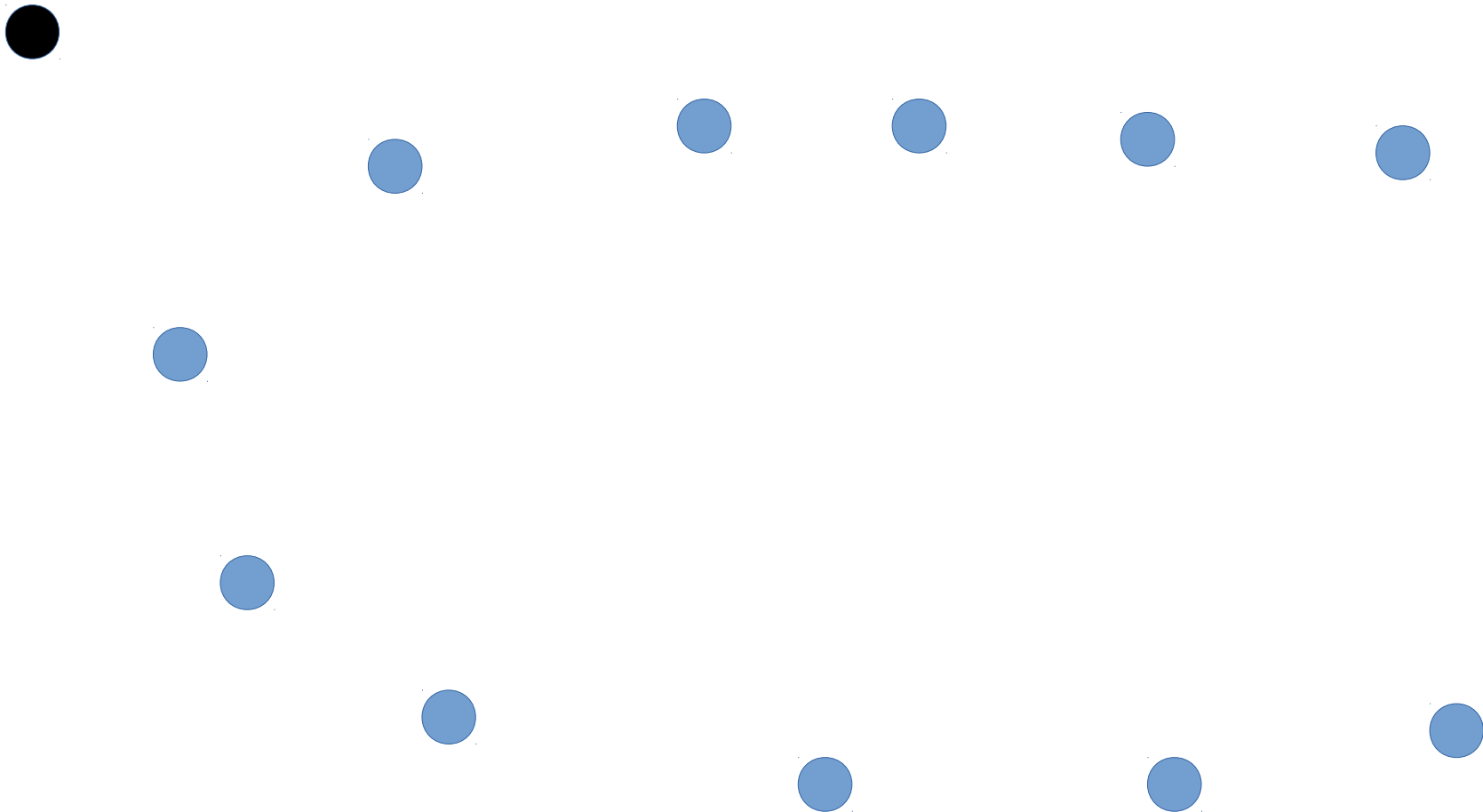


Définition par projection



« Moving Least Squares »

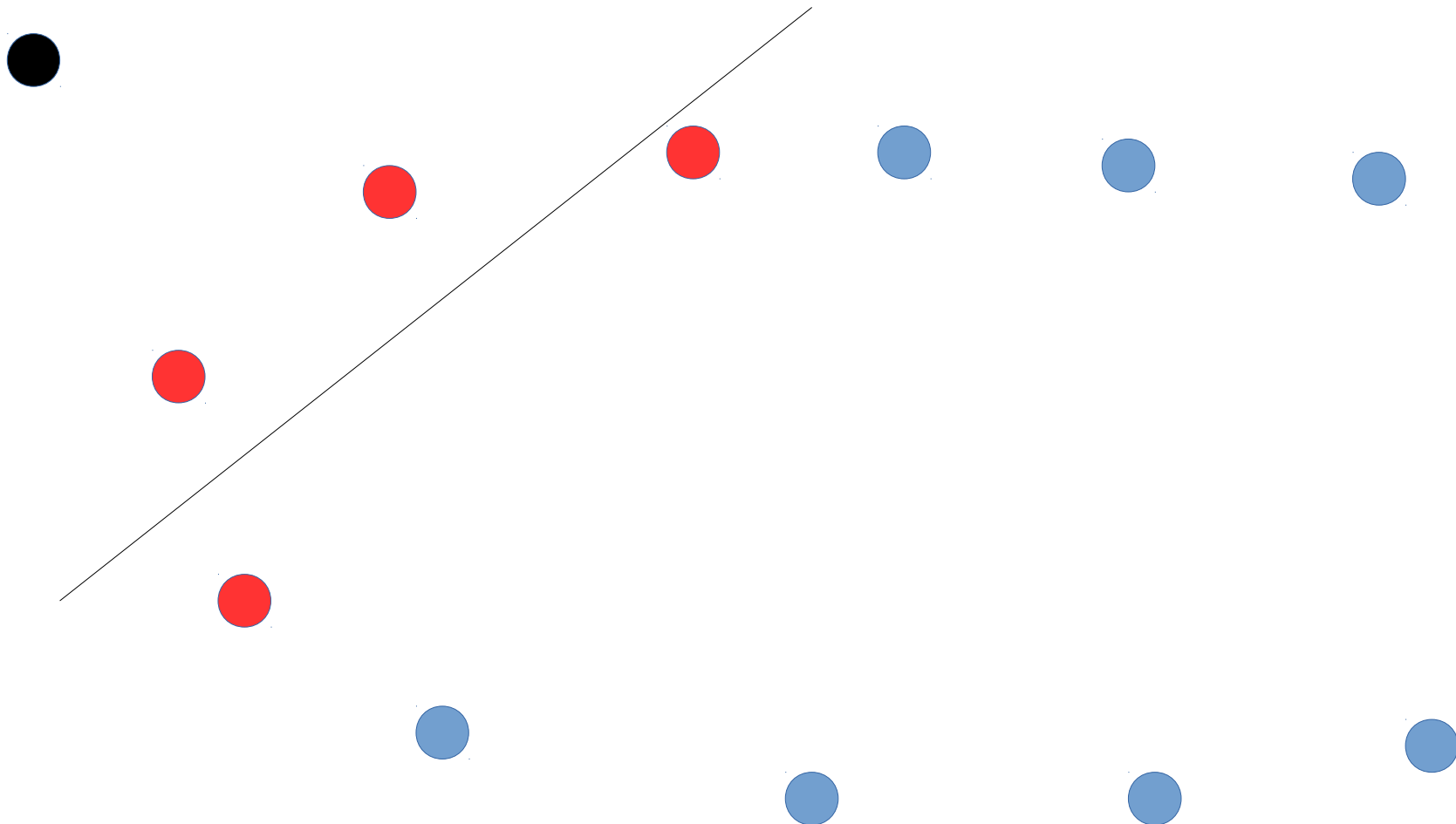
Stratégie de projection par MLS :



« Moving Least Squares »

Stratégie de projection par MLS :

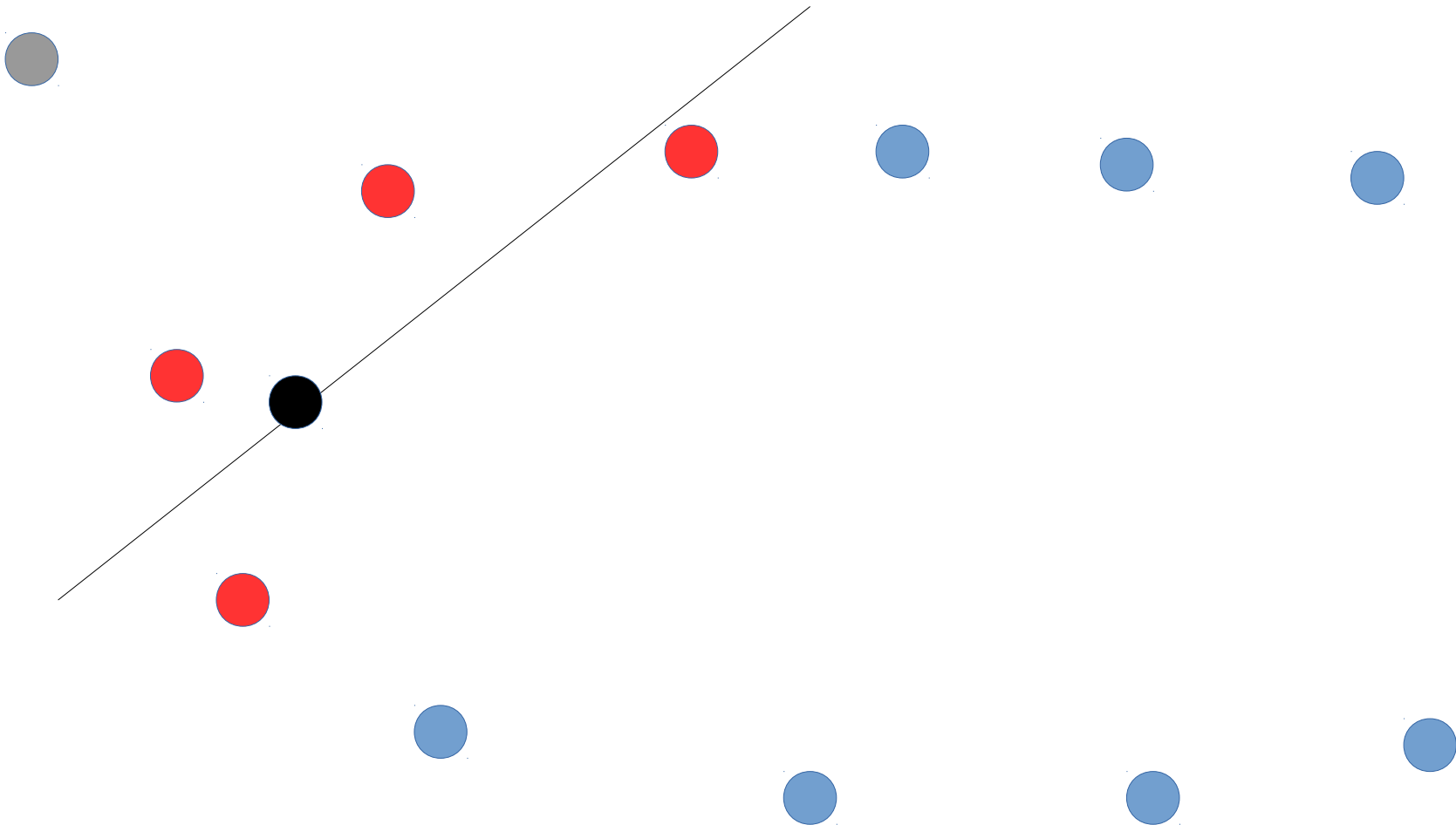
1) Trouver le plan qui approxime au mieux le voisinage



« Moving Least Squares »

Stratégie de projection par MLS :

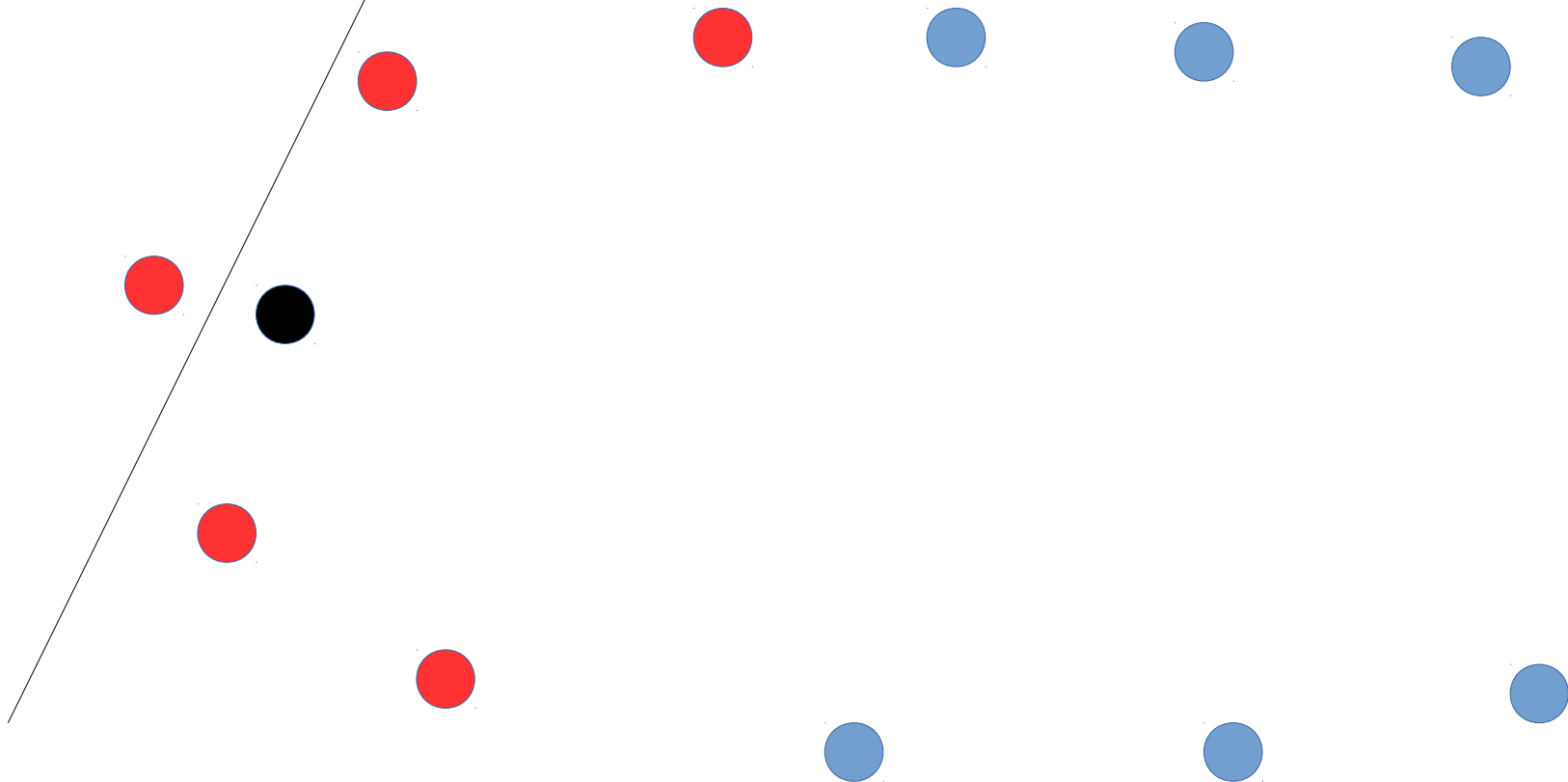
- 1) Trouver le plan qui approxime au mieux le voisinage
- 2) Projeter le point dessus



« Moving Least Squares »

Stratégie de projection par MLS :

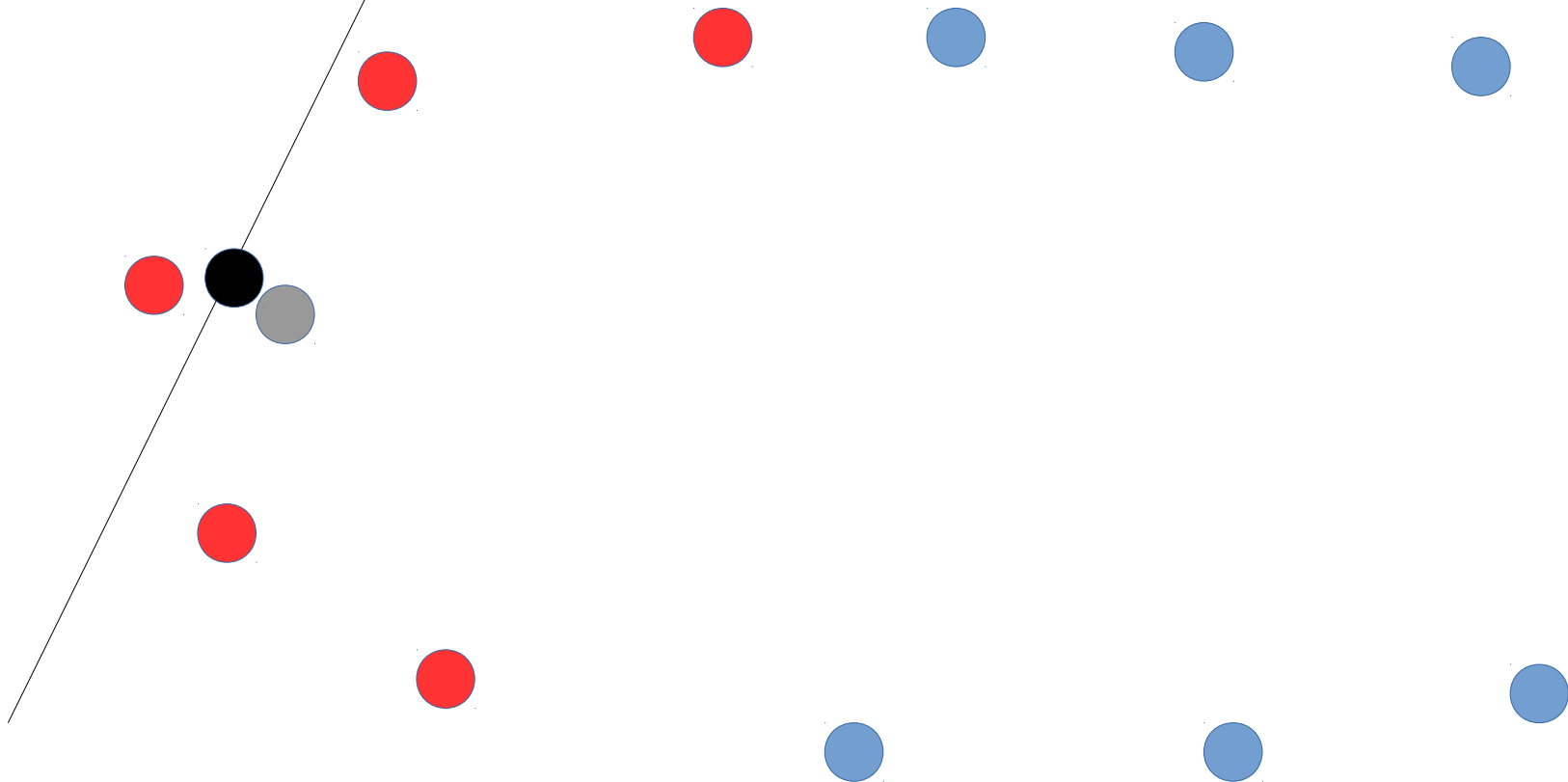
- 1) Trouver le plan qui approxime au mieux le voisinage
- 2) Projeter le point dessus
- 3) Recommencer (go to 1)



« Moving Least Squares »

Stratégie de projection par MLS :

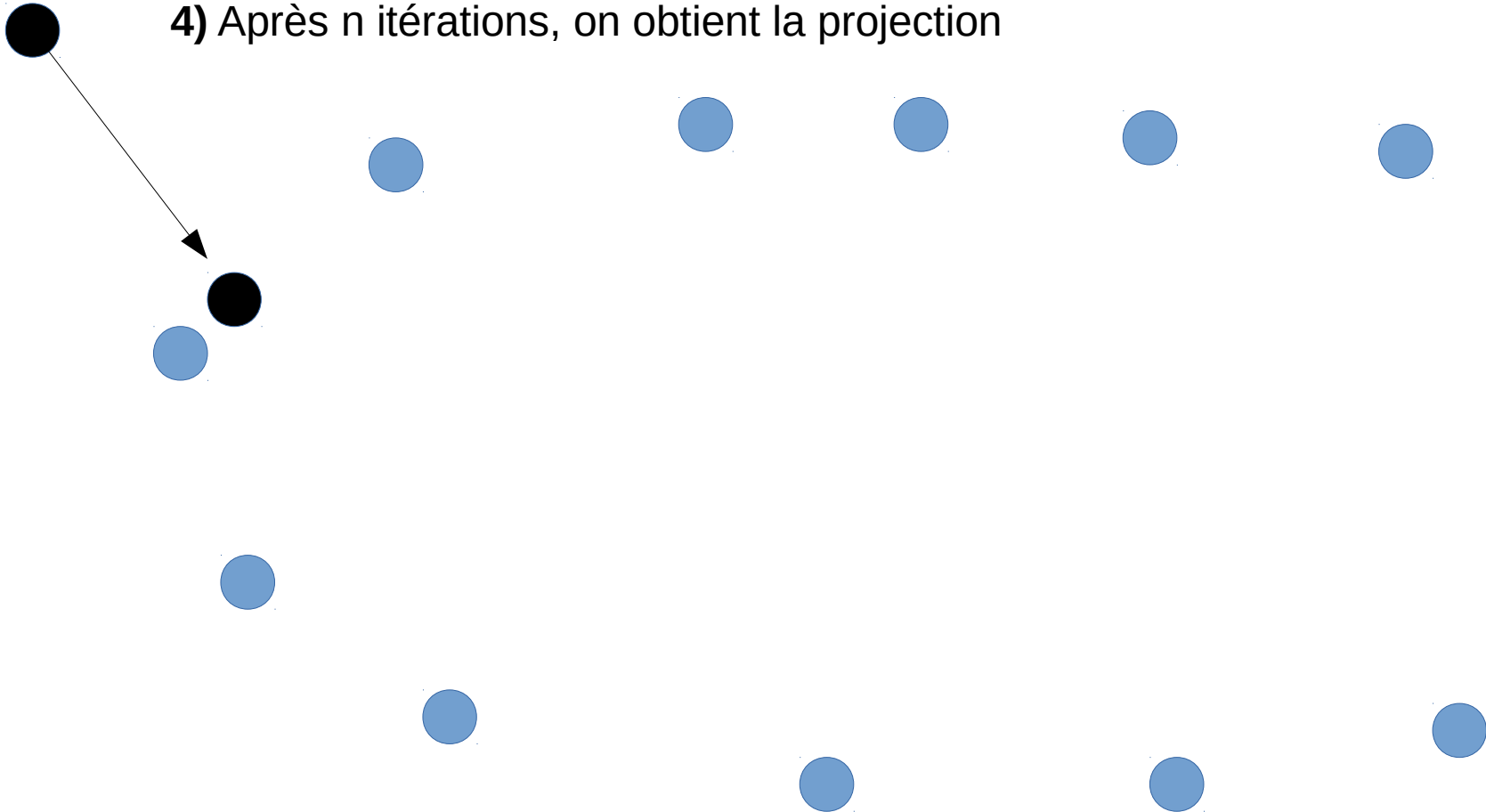
- 1) Trouver le plan qui approxime au mieux le voisinage
- 2) Projeter le point dessus
- 3) Recommencer (go to 1)



« Moving Least Squares »

Stratégie de projection par MLS :

- 1) Trouver le plan qui approxime au mieux le voisinage
- 2) Projeter le point dessus
- 3) Recommencer (go to 1)
- 4) Après n itérations, on obtient la projection



« Moving Least Squares »

- A chaque itération k , on cherche les plus proches voisins, et on définit des poids vis à vis d'eux (les points lointains sont donc ignorés)
- On trouve le plan qui passe au mieux par ces points (ACP pondérée)

$$\left\{ \begin{array}{l} c = \sum_{p_i \in NN(x)} w_i p_i / \sum_{p_i \in NN(x)} w_i \\ C = \sum_{p_i \in NN(x)} w_i (p_i - c) \cdot (p_i - c)^T \in \mathbb{R}^{3 \times 3} \end{array} \right. \quad n = eig_3(C)$$
- On projette sur ce plan :
$$x_{k+1} = x_k - \frac{(x_k - c)^T \cdot n}{\|n\|^2} n$$
- « Moving Least Squares » car les poids varient en fonction de la position du point x_k au fil des itérations $\{k\}$, et qu'on approxime « aux moindres carrés » la géométrie voisine.

Définition des poids par noyau

x



$$w_i = \phi(\|x - p_i\|)$$

:Influence du point
d'entrée sur x

p_i



Définition des poids par noyau

x



$$w_i = \phi(\|x - p_i\|)$$

:Influence du point
d'entrée sur x

p_i



$$\phi(r) = e^{\frac{-r^2}{h^2}}$$

Définition des poids par noyau

x



$$w_i = \phi(\|x - p_i\|)$$

:Influence du point
d'entrée sur x

p_i



$$\phi(r) = e^{\frac{-r^2}{h^2}}$$

$$\phi(r) = \left(1 - \frac{r}{h}\right)^4 \cdot \left(1 + 4\frac{r}{h}\right)$$

Définition des poids par noyau

x



$$w_i = \phi(\|x - p_i\|)$$

:Influence du point
d'entrée sur x

p_i



$$\phi(r) = e^{\frac{-r^2}{h^2}}$$

$$\phi(r) = \left(1 - \frac{r}{h}\right)^4 \cdot \left(1 + 4 \frac{r}{h}\right)$$

$$\phi(r) = \left(\frac{r}{h}\right)^{-w}$$

Définition des poids par noyau

x



$$w_i = \phi(\|x - p_i\|)$$

:Influence du point d'entrée sur x

p_i



$$\phi(r) = e^{\frac{-r^2}{h^2}}$$

$$\phi(r) = \left(1 - \frac{r}{h}\right)^4 \cdot \left(1 + 4 \frac{r}{h}\right)$$

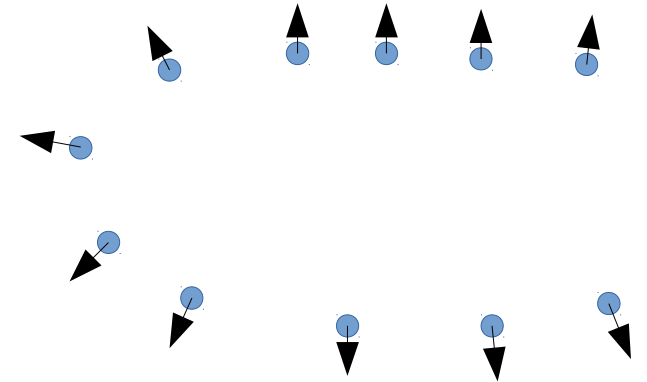
$$\phi(r) = \left(\frac{r}{h}\right)^{-w}$$

→ Surface **approximant** les points d'entrée

→ Surface **interpolant** les points d'entrée

« Point set surfaces »

- Modernes définitions de surfaces de points considèrent que le nuage d'entrée doit être équipé de normales.
- On utilise
$$\begin{cases} c(x) = \sum_{p_i \in NN(x)} w_i p_i / \sum_{p_i \in NN(x)} w_i & \text{à la place de la PCA.} \\ n(x) = \sum_{p_i \in NN(x)} w_i n_i / \left\| \sum_{p_i \in NN(x)} w_i n_i \right\| \end{cases}$$

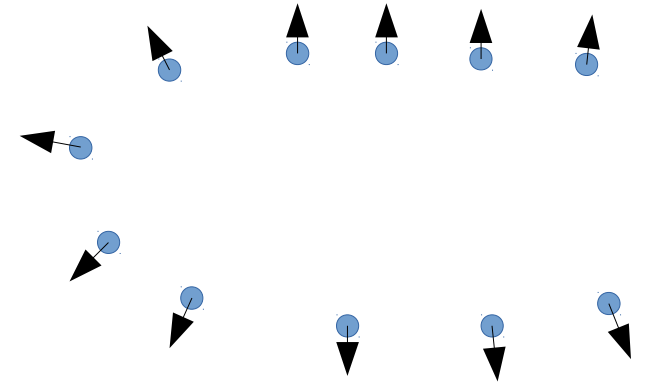


« Point set surfaces »

- Modernes définitions de surfaces de points considèrent que le nuage d'entrée doit être équipé de normales.

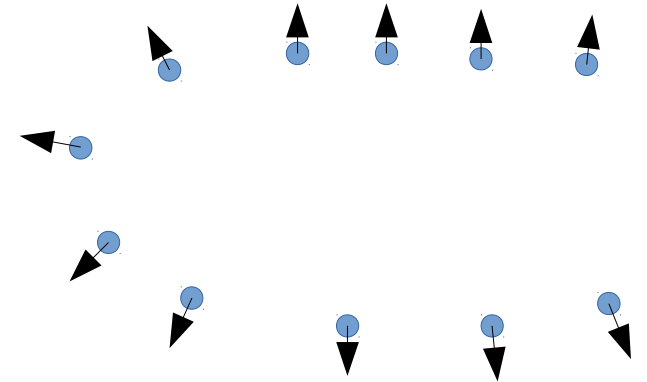
- On utilise
$$\begin{cases} c(x) = \sum_{p_i \in NN(x)} w_i p_i / \sum_{p_i \in NN(x)} w_i \\ n(x) = \sum_{p_i \in NN(x)} w_i n_i / \left\| \sum_{p_i \in NN(x)} w_i n_i \right\| \end{cases}$$
 a la place de la PCA.

- On peut définir une fonction implicite $f(x) = (x - c(x))^T \cdot n(x)$ (la surface est alors le « 0-set » de cette fonction).

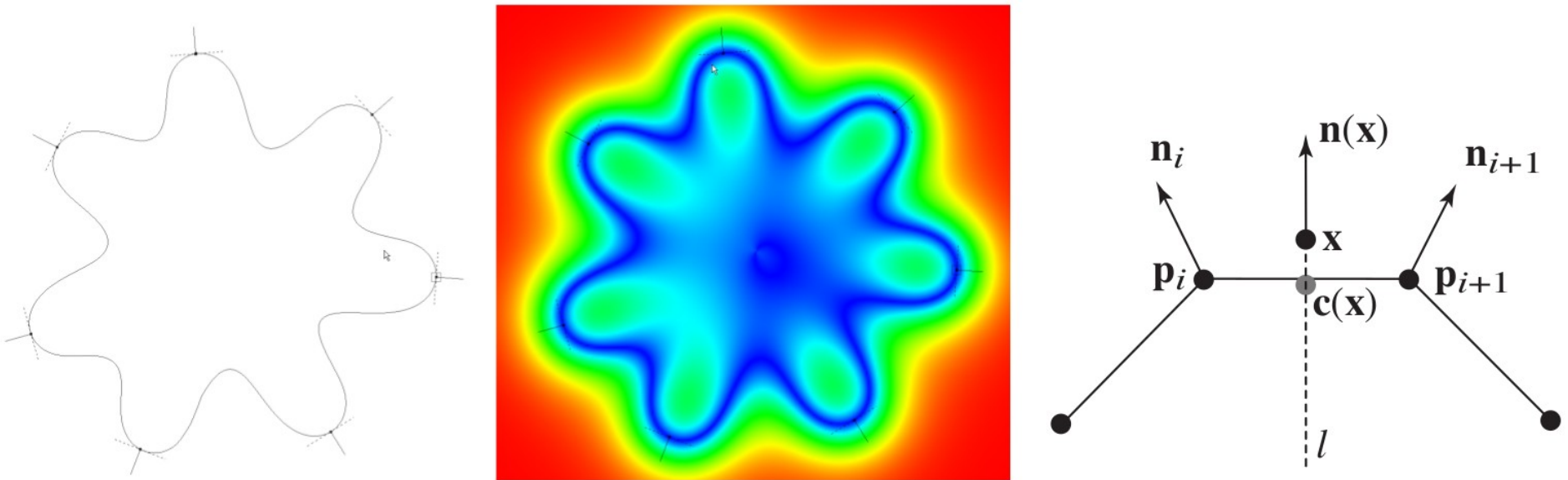


« Point set surfaces »

- Modernes définitions de surfaces de points considèrent que le nuage d'entrée doit être équipé de normales.
- On utilise
$$\begin{cases} c(x) = \sum_{p_i \in NN(x)} w_i p_i / \sum_{p_i \in NN(x)} w_i \\ n(x) = \sum_{p_i \in NN(x)} w_i n_i / \left\| \sum_{p_i \in NN(x)} w_i n_i \right\| \end{cases}$$
 a la place de la PCA.
- On peut définir une fonction implicite $f(x) = (x - c(x))^T \cdot n(x)$ (la surface est alors le « 0-set » de cette fonction).
- Avec un noyau très singulier, on s'attend à pouvoir interpoler points ET normales.
- Ce schéma de PSS est généralement appelé SPSS (Simple Point Set Surface)

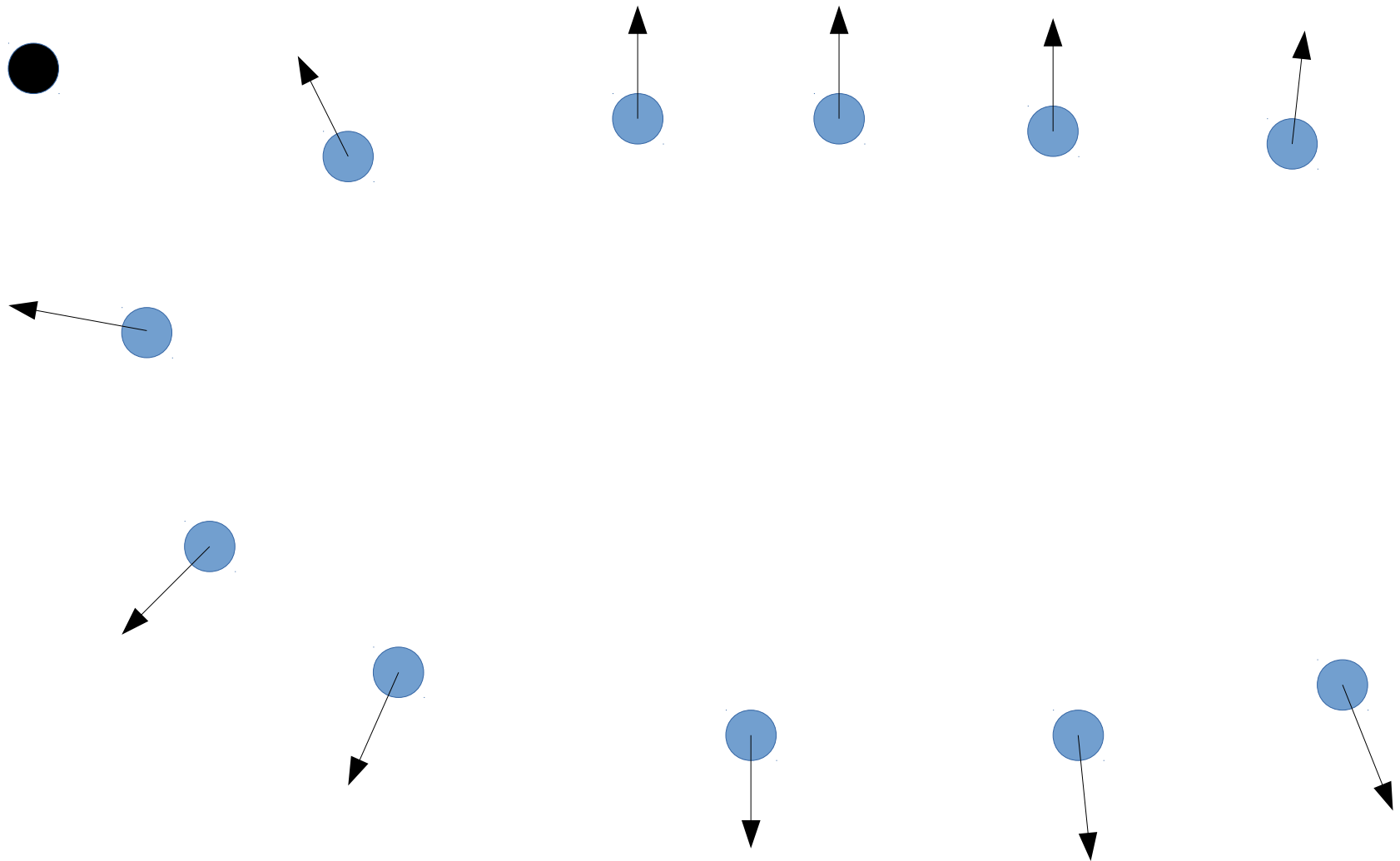


« Point set surfaces » : problemes du schéma standard



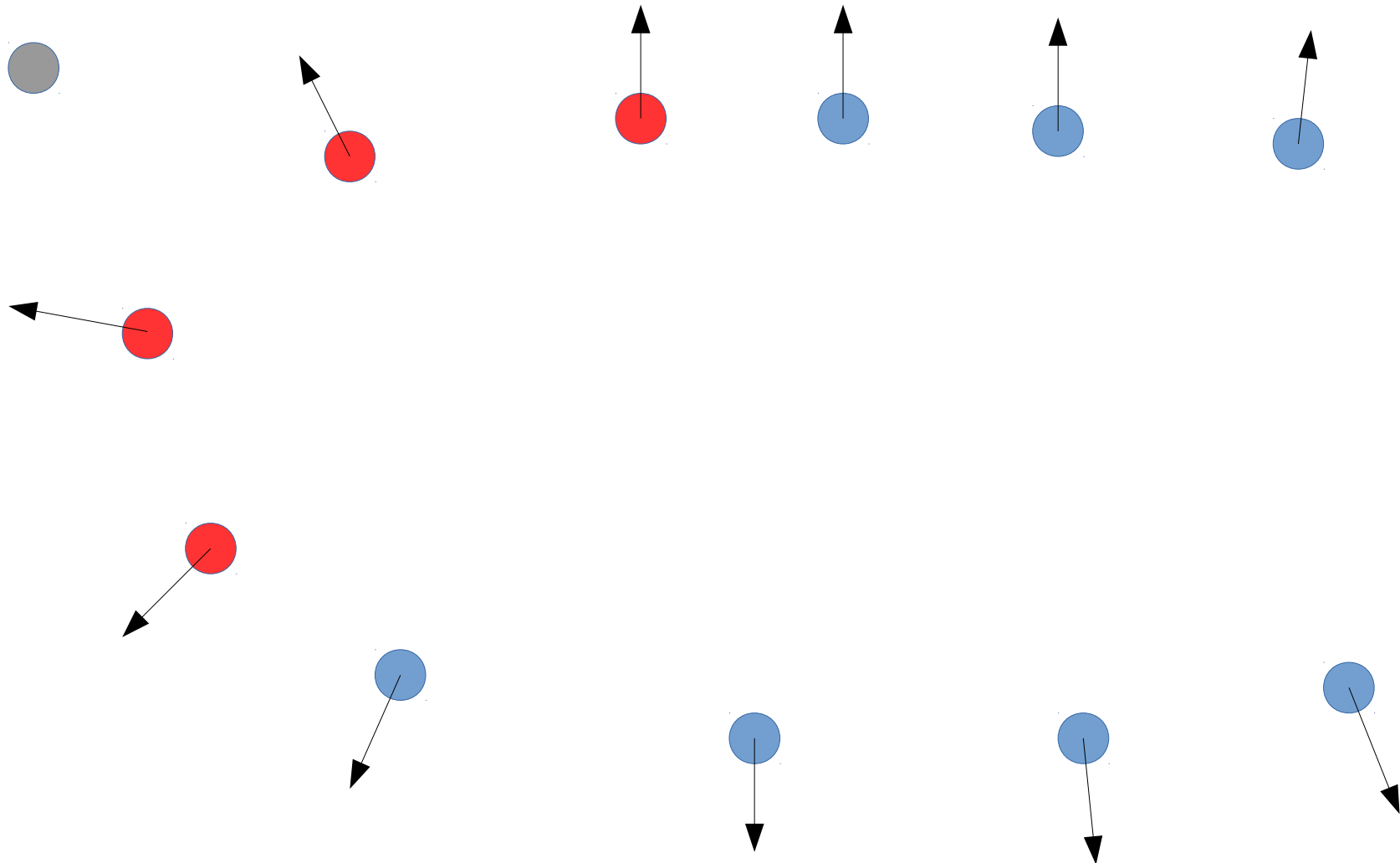
Impossible d'obtenir des surfaces interpolantes convexes.

« Hermite point set surfaces »



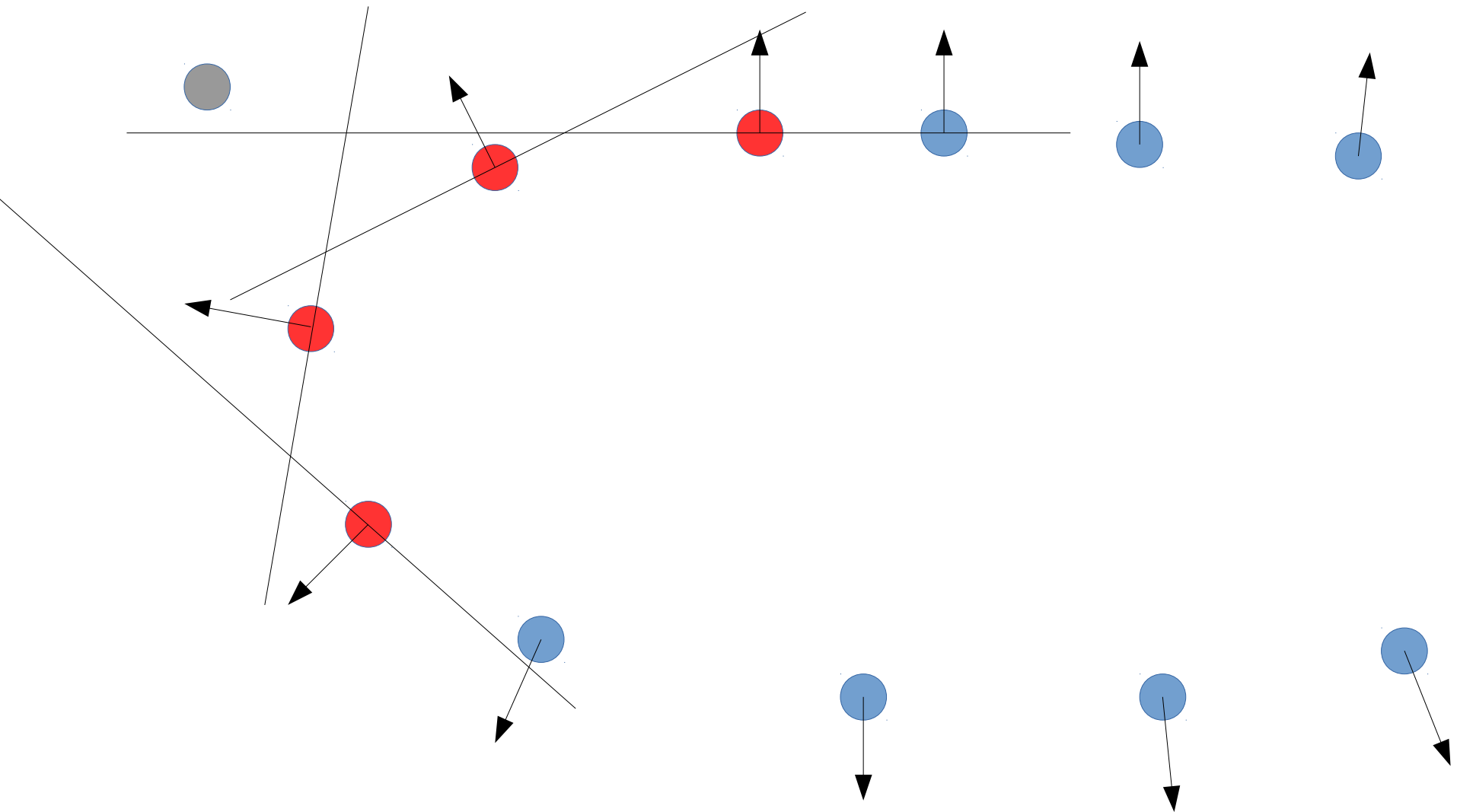
« Hermite point set surfaces »

On trouve les plus proches voisins



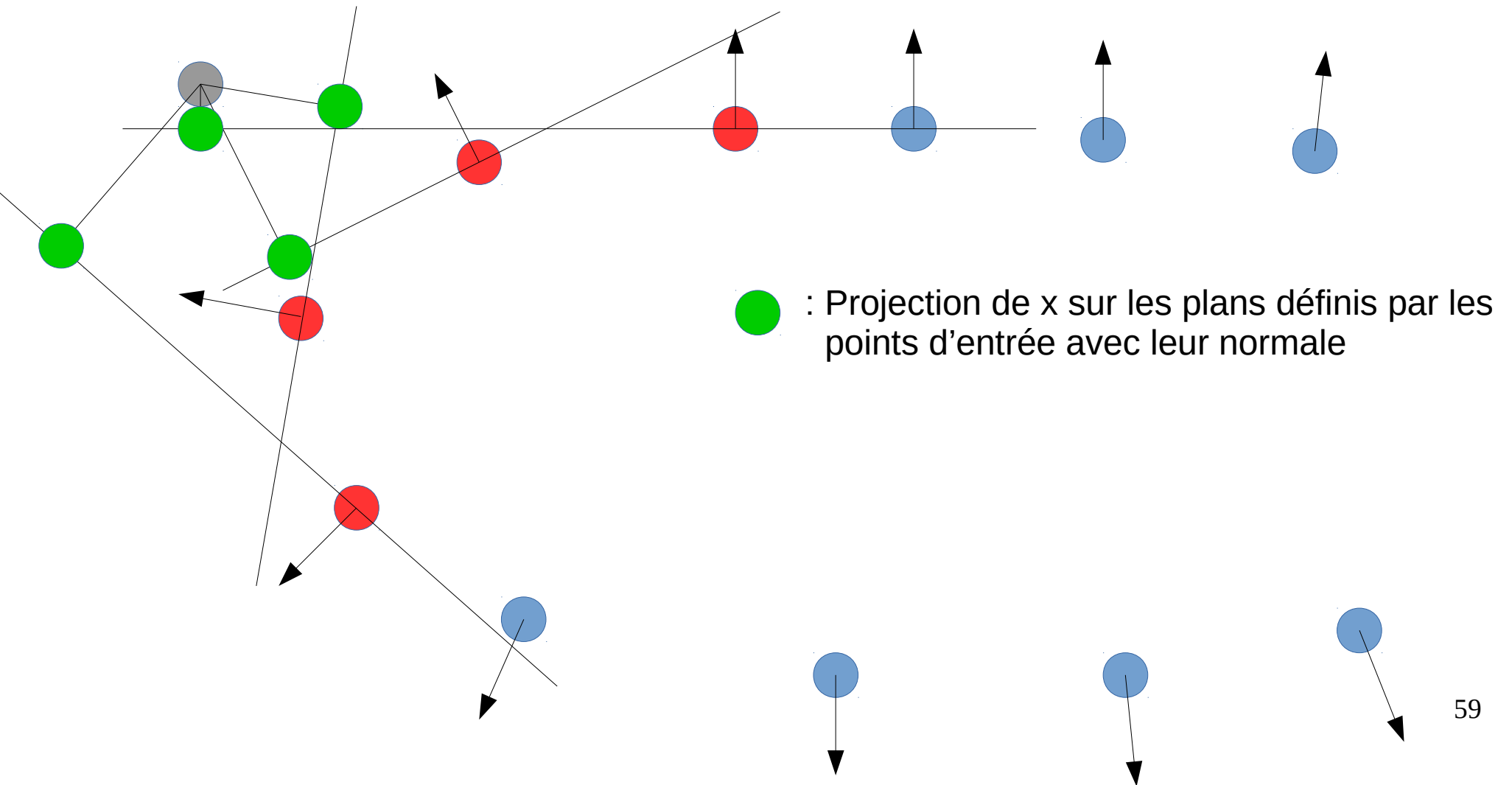
« Hermite point set surfaces »

On considère leurs **plans tangents**



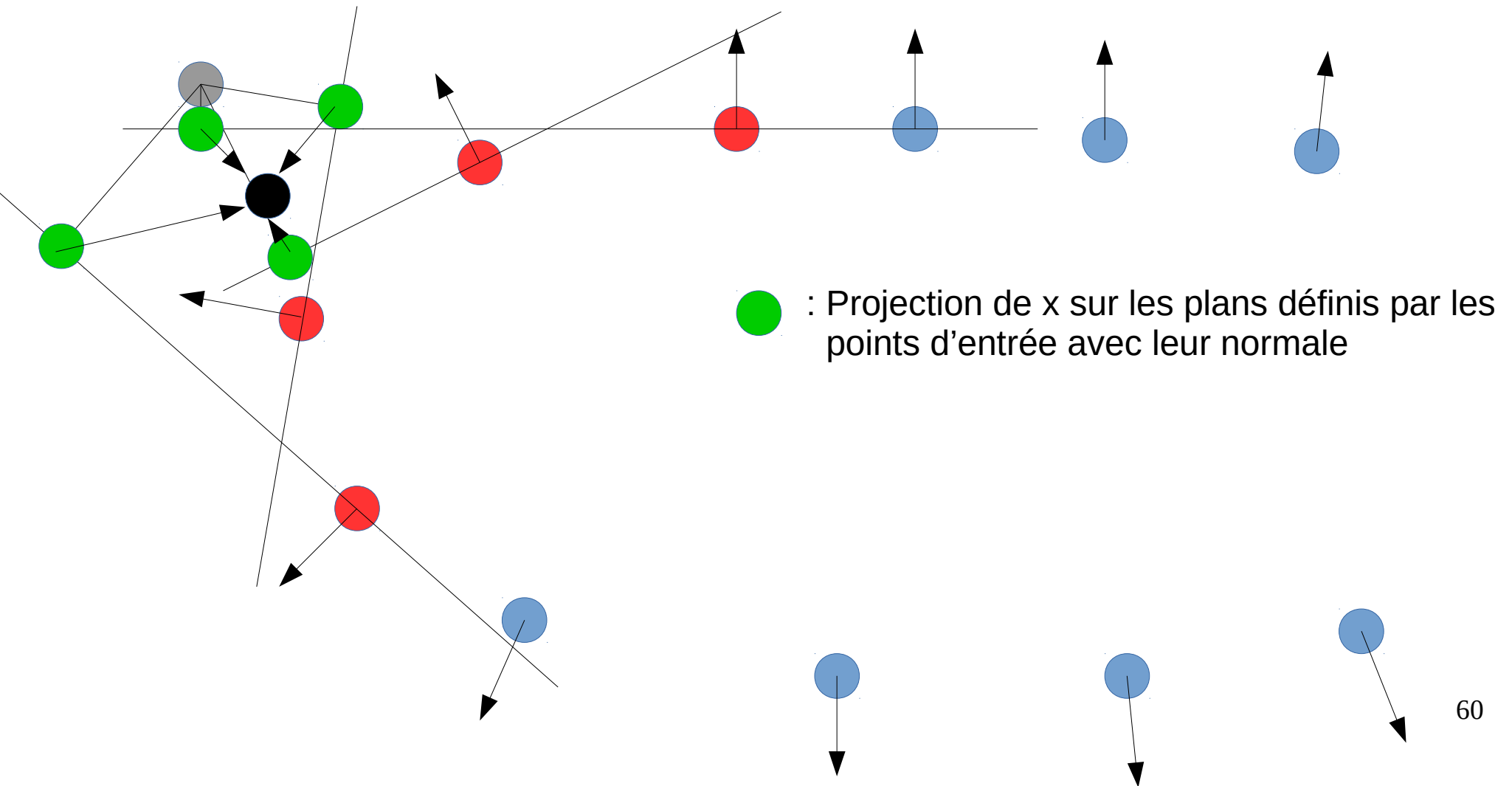
« Hermite point set surfaces »

On projette le point sur ces plans tangents



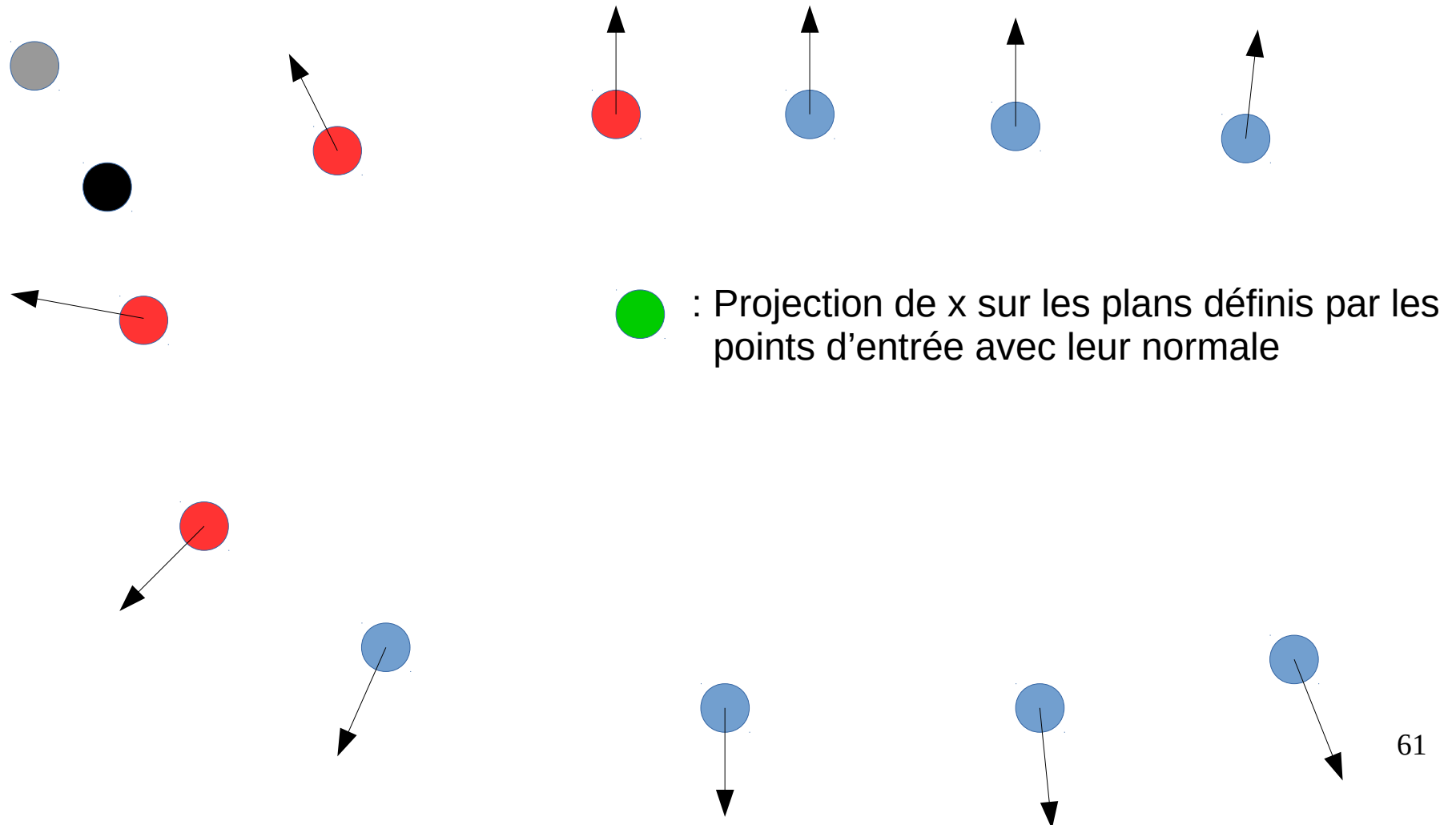
« Hermite point set surfaces »

On moyenne **ces points projetés** plutôt que les points d'entrée pour calculer le centroïde



« Hermite point set surfaces »

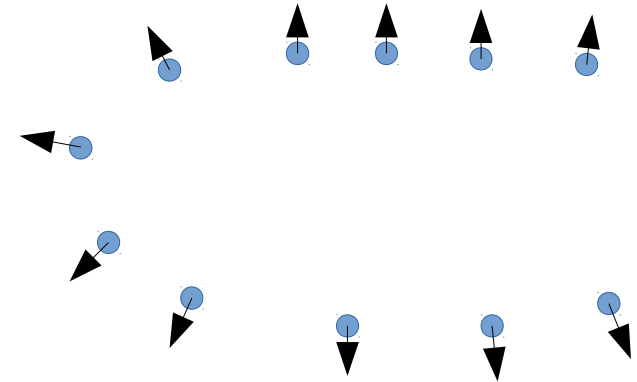
On peut maintenant considérer un plan de projection dont le centroïde n'est pas nécessairement à l'intérieur de l'enveloppe convexe des points d'entrée.



« Hermite point set surfaces »

- On utilise

$$\left\{ \begin{array}{l} \tilde{p}_i(x) = x - ((x - p_i)^T \cdot n_i) n_i \\ c(x) = \sum_{p_i \in NN(x)} w_i \tilde{p}_i(x) / \sum_{p_i \in NN(x)} w_i \\ n(x) = \sum_{p_i \in NN(x)} w_i n_i / \left\| \sum_{p_i \in NN(x)} w_i n_i \right\| \end{array} \right.$$



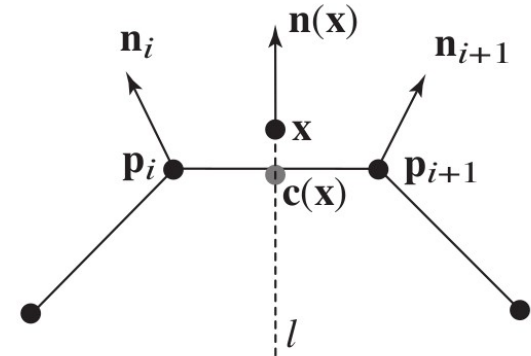
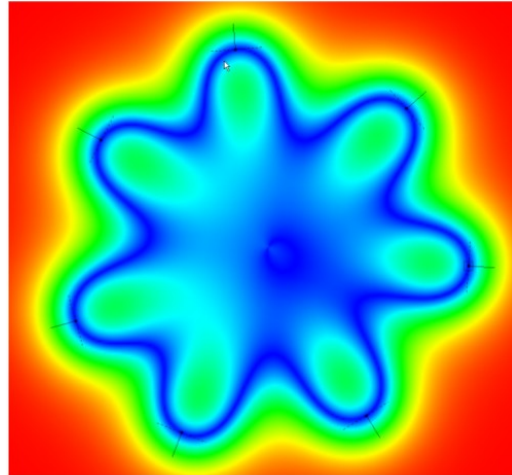
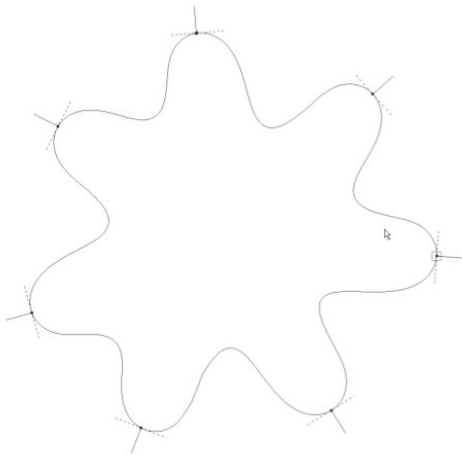
- Schémas populaires de projection itérative :

- Simple : $x_{k+1} = \text{project}(x_k, (c(x_k), n(n_k)))$

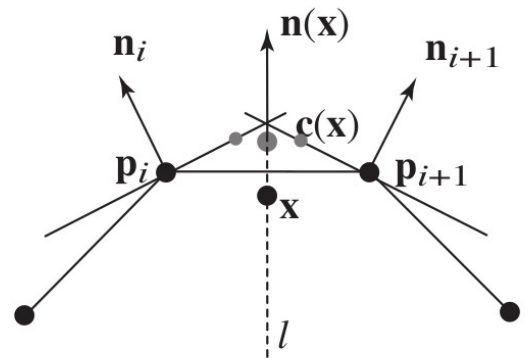
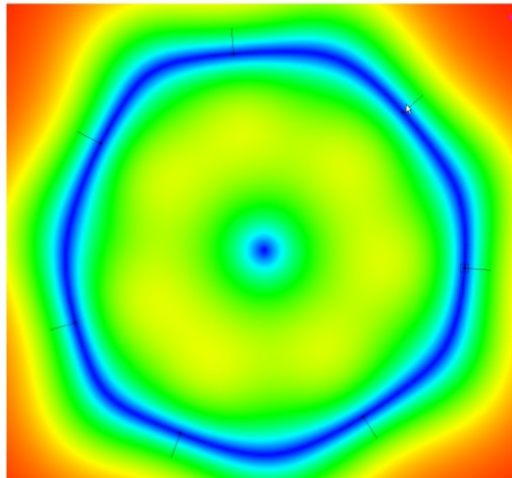
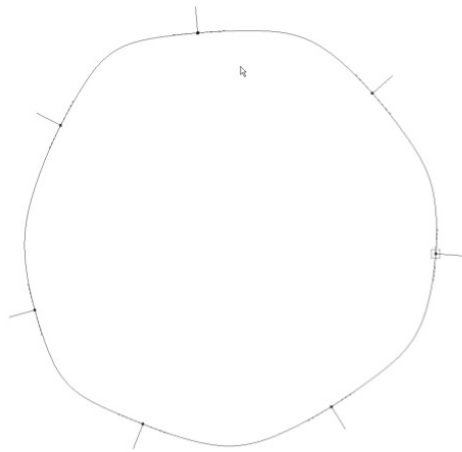
- « Presque orthogonal » : $x_{k+1} = \text{project}(x, (c(x_k), n(n_k)))$

« Hermite Point set surfaces »

PSS



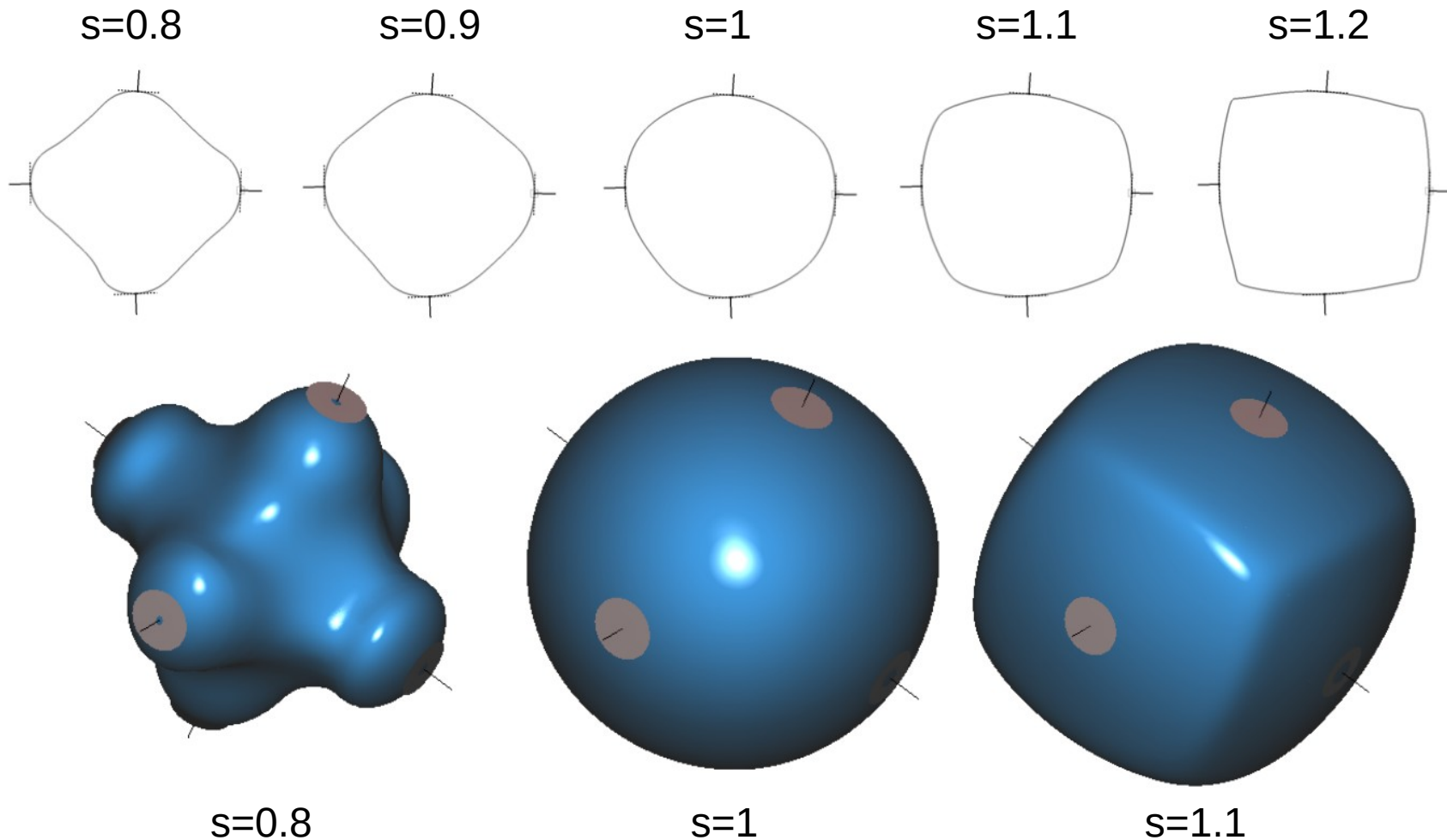
HPSS



$$f(x) = (x - c(x))^T \cdot n(x)$$

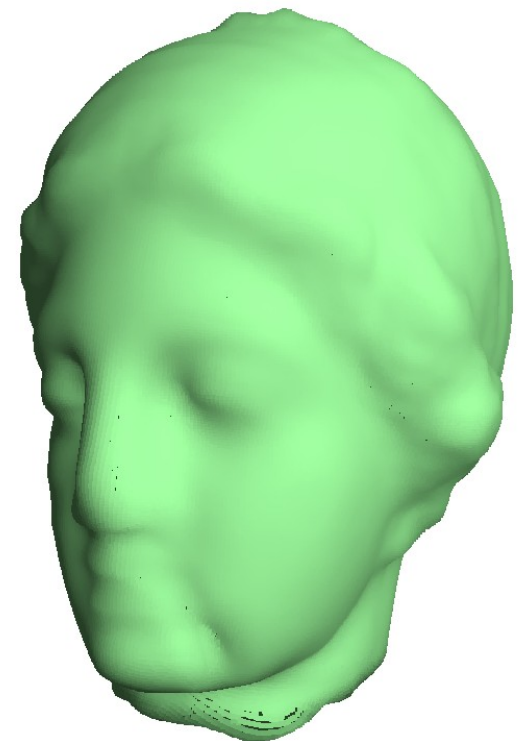
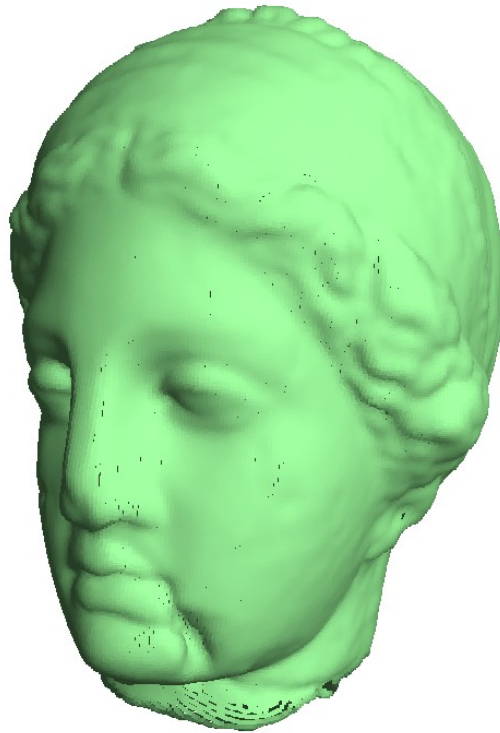
« Hermite Point set surfaces »

Extension : $\tilde{p}_i(x, s) = s\tilde{p}_i(x) + (1-s)p_i$

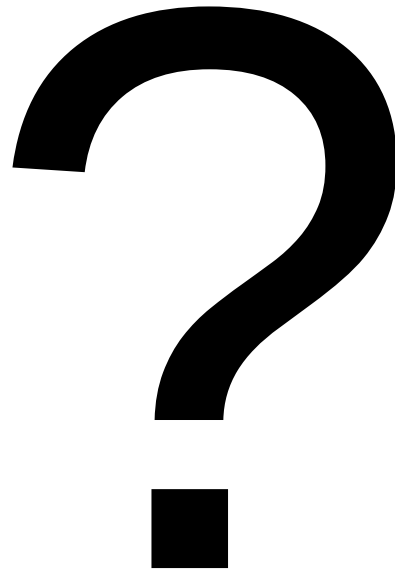


« Hermite Point set surfaces »

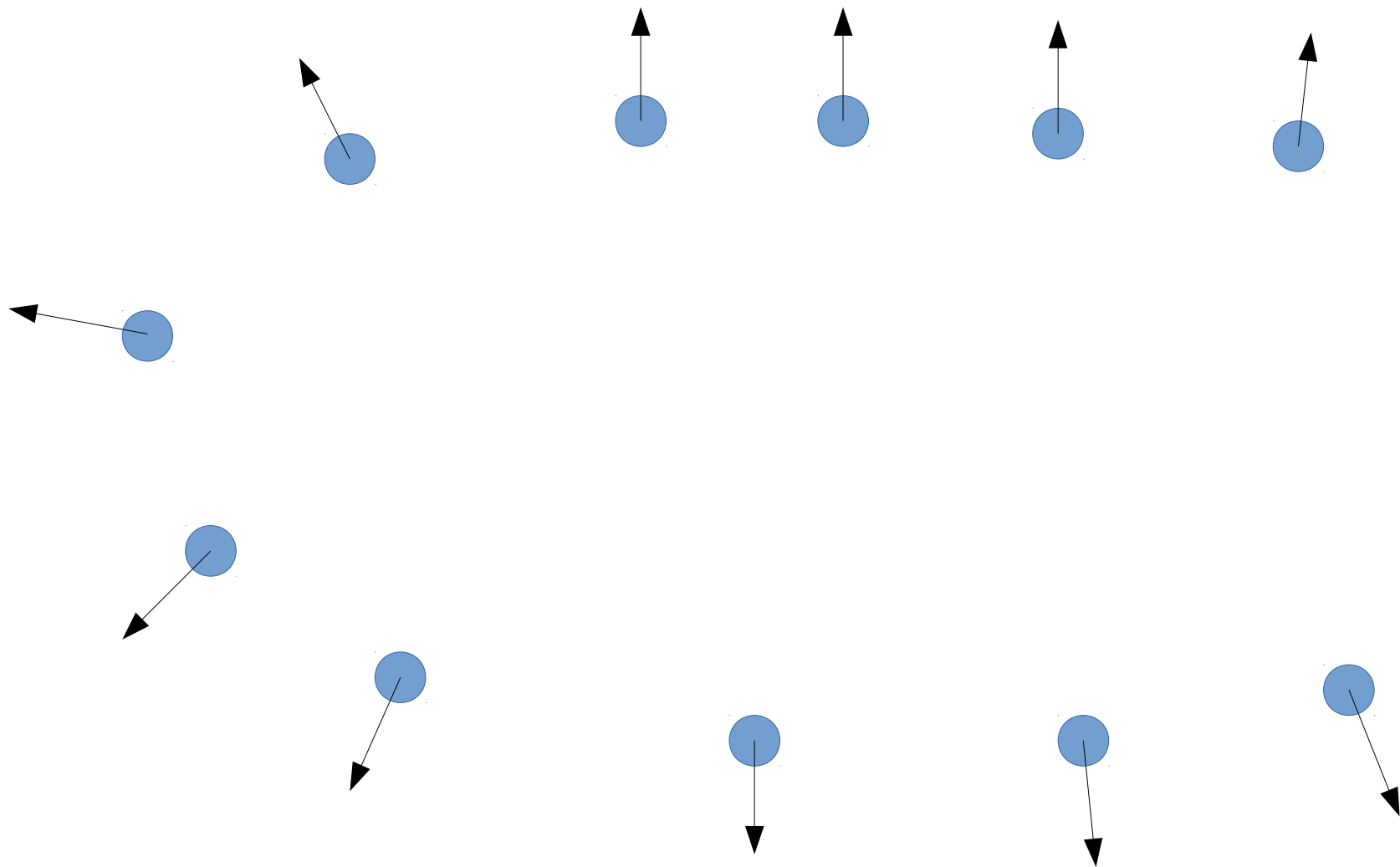
Application au filtrage, analogie avec le filtrage Gaussien d'images



Questions ?

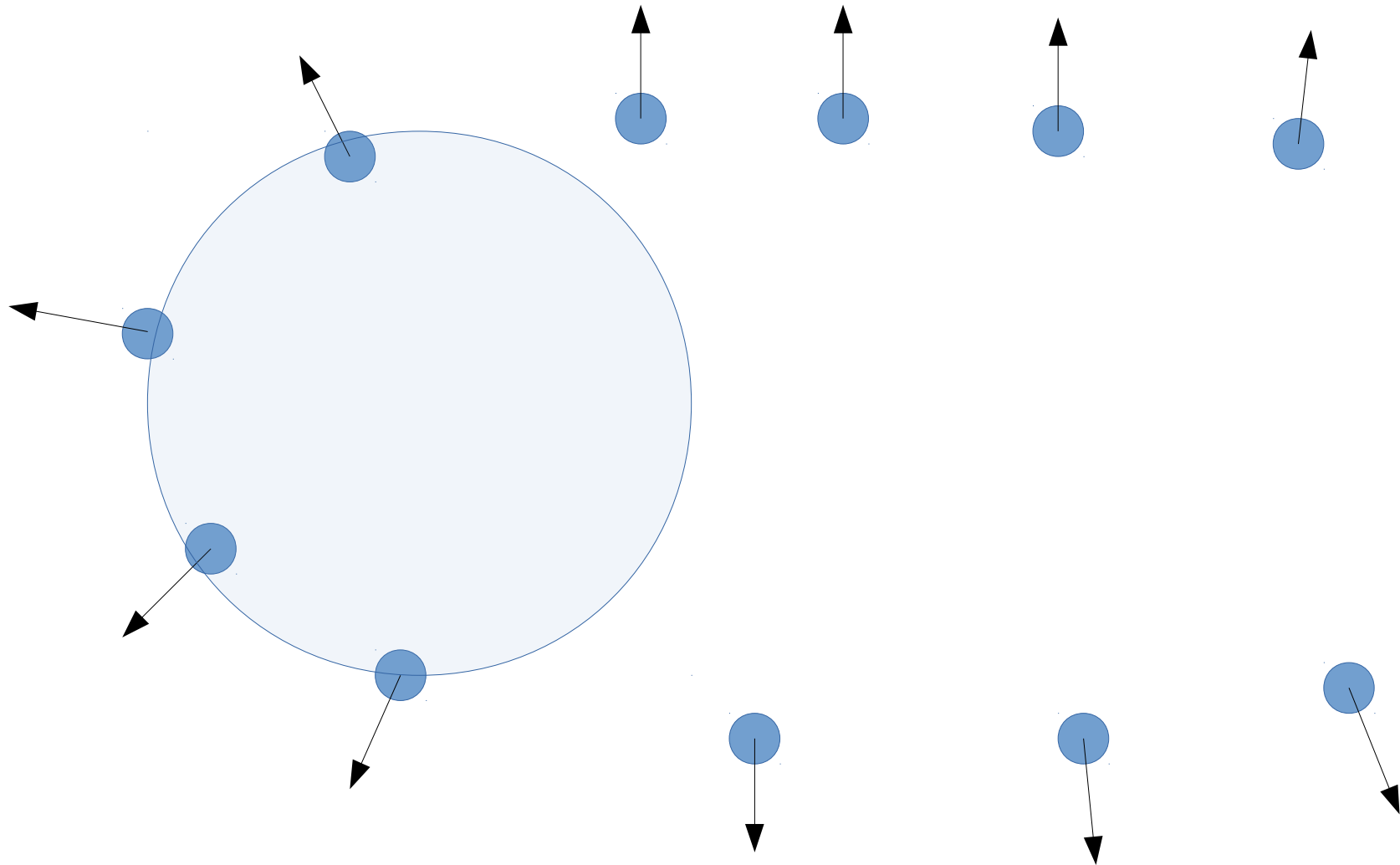


Surface de points algébrique



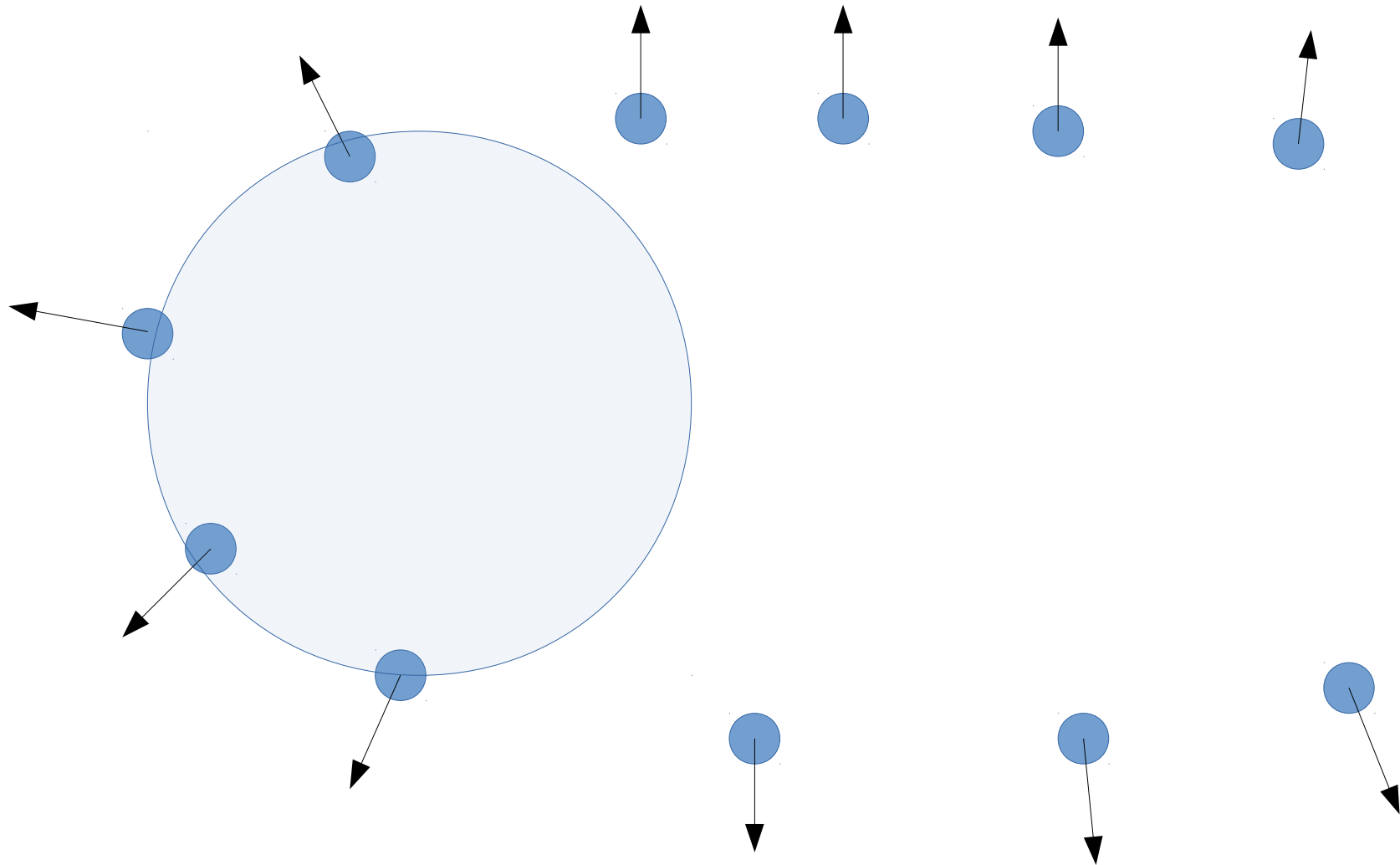
Surface de points algébrique

- Idée principale : projeter sur des sphères plutôt que des plans



Surface de points algébrique

- Idée principale : projeter sur des sphères plutôt que des plans
- « Algébrique », car une sphère dégénère mal vers un plan (le centre est à l'infini...), mais **son équation algébrique dégénère continûment vers celle du plan...**



Surface de points algébrique

- Idée principale : projeter sur des sphères plutôt que des plans
- « Algébrique », car une sphère dégénère mal vers un plan (le centre est à l'infini...), mais **son équation algébrique dégénère continûment vers celle du plan...**

$$\|X - c\|^2 - r^2 = 0$$

Eq standard d'une sphère

$$(\|c\|^2 - r^2) + X^T \cdot (-2c) + \|X\|^2 = 0$$

Eq revisitée

Surface de points algébrique

- Idée principale : projeter sur des sphères plutôt que des plans
- « Algébrique », car une sphère dégénère mal vers un plan (le centre est à l'infini...), mais **son équation algébrique dégénère continûment vers celle du plan...**

$$\|X - c\|^2 - r^2 = 0$$

Eq standard d'une sphère

$$(\|c\|^2 - r^2) + X^T \cdot (-2c) + \|X\|^2 = 0$$

Eq revisitée

$$d + n^T \cdot X = 0$$

Eq standard d'un plan

Surface de points algébrique

- Idée principale : projeter sur des sphères plutôt que des plans
- « Algébrique », car une sphère dégénère mal vers un plan (le centre est à l'infini...), mais **son équation algébrique dégénère continûment vers celle du plan...**

$$\|X - c\|^2 - r^2 = 0$$

Eq standard d'une sphère

$$(\|c\|^2 - r^2) + X^T \cdot (-2c) + \|X\|^2 = 0$$

Eq revisitée

$$d + n^T \cdot X = 0$$

Eq standard d'un plan

$$(1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T = 0$$

Eq générale d'une sphère algébrique

Surface de points algébrique

$$(1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T = f(X)$$

Eq générale d'une sphère algébrique

$$f(X) = 0 \quad \text{Pour les points sur la sphère}$$

$$\nabla f(X) \quad \text{Normale pour X sur la sphère}$$

$$\nabla f(X) = u_{123} + 2u_4 X$$

Surface de points algébrique

$$(1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T = f(X)$$

Eq générale d'une sphère algébrique

$$f(X) = 0 \quad \text{Pour les points sur la sphère}$$

$$\nabla f(X) \quad \text{Normale pour X sur la sphère}$$

$$\nabla f(X) = u_{123} + 2u_4 X$$

Fitting en deux temps d'une sphère a un ensemble de points : $\{w_i, p_i, n_i\}$

1) Minimiser $\sum_i w_i \|\nabla f(p_i) - n_i\|^2$ (cela définit u_{123} , et u_4)

2) Minimiser $\sum_i w_i (f(p_i))^2$ (cela définit u_0)

Stratégie adoptée par :

[Guennebaud et al. 2008] *Dynamic Sampling and Rendering of APSS*

Minimiser l'équation normale

1) Minimiser $\sum_i w_i \|\nabla f(p_i) - n_i\|^2$ (cela définit u_{123} , et u_4)

$$f(X) = (1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T \longrightarrow \nabla f(X) = (I_3, 2X) \cdot (u_{123}, u_4)^T = (I_3, 2X) \cdot U$$

Minimiser l'équation normale

1) Minimiser $\sum_i w_i \|\nabla f(p_i) - n_i\|^2$ (cela définit u_{123} , et u_4)

$$f(X) = (1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T \longrightarrow \nabla f(X) = (I_3, 2X) \cdot (u_{123}, u_4)^T = (I_3, 2X) \cdot U$$

Or
$$\sum_i w_i \|\nabla f(p_i) - n_i\|^2 = \sum_i w_i \nabla f(p_i)^T \cdot \nabla f(p_i) - 2 \sum_i w_i \nabla f(p_i)^T \cdot n_i + \text{const}$$

Minimiser l'équation normale

1) Minimiser $\sum_i w_i \|\nabla f(p_i) - n_i\|^2$ (cela définit u_{123} , et u_4)

$$f(X) = (1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T \longrightarrow \nabla f(X) = (I_3, 2X) \cdot (u_{123}, u_4)^T = (I_3, 2X) \cdot U$$

Or
$$\sum_i w_i \|\nabla f(p_i) - n_i\|^2 = \sum_i w_i \nabla f(p_i)^T \cdot \nabla f(p_i) - 2 \sum_i w_i \nabla f(p_i)^T \cdot n_i + \text{const}$$

\longrightarrow Minimiser
$$\sum_i w_i \nabla f(p_i)^T \cdot \nabla f(p_i) - 2 \sum_i w_i \nabla f(p_i)^T \cdot n_i$$

qui vaut
$$U^T \cdot \left(\sum_i w_i (I_3, 2p_i)^T \cdot (I_3, 2p_i) \right) \cdot U - 2 \left(\sum_i w_i (I_3, 2p_i)^T \cdot n_i \right)^T \cdot U$$

Minimiser l'équation normale

1) Minimiser $\sum_i w_i \|\nabla f(p_i) - n_i\|^2$ (cela définit u_{123} , et u_4)

$$f(X) = (1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T \longrightarrow \nabla f(X) = (I_3, 2X) \cdot (u_{123}, u_4)^T = (I_3, 2X) \cdot U$$

Or $\sum_i w_i \|\nabla f(p_i) - n_i\|^2 = \sum_i w_i \nabla f(p_i)^T \cdot \nabla f(p_i) - 2 \sum_i w_i \nabla f(p_i)^T \cdot n_i + \text{const}$

→ Minimiser $\sum_i w_i \nabla f(p_i)^T \cdot \nabla f(p_i) - 2 \sum_i w_i \nabla f(p_i)^T \cdot n_i$

qui vaut $U^T \cdot \left(\sum_i w_i (I_3, 2p_i)^T \cdot (I_3, 2p_i) \right) \cdot U - 2 \left(\sum_i w_i (I_3, 2p_i)^T \cdot n_i \right)^T \cdot U$

→ Quadrique en U (le minimum est donné par $A^{-1} \cdot B$)

si $Q(U) = U^T \cdot A \cdot U - 2B^T \cdot U + C$

Minimiser l'équation de position

1) Minimiser $\sum_i w_i (f(p_i))^2$ (cela définit u_0)

$$f(X) = (1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T$$

$$f(p_i) = u_0 + (p_i^T, \|p_i\|^2) \cdot (u_{123}, u_4)^T = u_0 + \alpha_i$$

→ Minimiser $\sum_i w_i (u_0 + \alpha_i)^2$

Minimiser l'équation de position

1) Minimiser $\sum_i w_i (f(p_i))^2$ (cela définit u_0)

$$f(X) = (1, X^T, \|X\|^2) \cdot (u_0, u_{123}, u_4)^T$$

$$f(p_i) = u_0 + (p_i^T, \|p_i\|^2) \cdot (u_{123}, u_4)^T = u_0 + \alpha_i$$

→ Minimiser $\sum_i w_i (u_0 + \alpha_i)^2$

→ Simple barycentre : $u_0 = - \sum_i w_i \alpha_i / \sum_i w_i$

Solution exacte

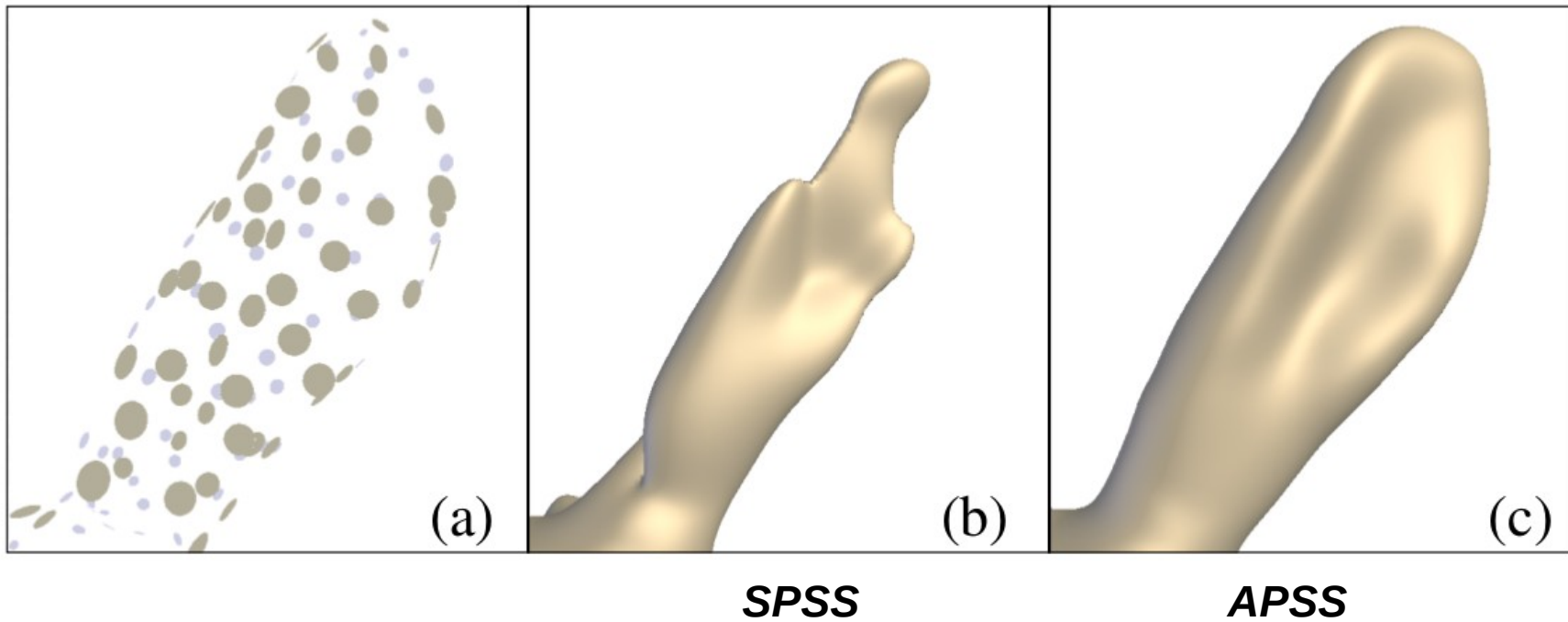
$$\left\{ \begin{array}{l} u_4 = \frac{1}{2} \frac{(\sum_i w_i p_i^T \cdot n_i) - (\sum_i w_i p_i)^T \cdot (\sum_j w_j n_j) / (\sum_k w_k)}{(\sum_i w_i p_i^T \cdot p_i) - (\sum_i w_i p_i)^T \cdot (\sum_j w_j p_j) / (\sum_k w_k)} \\ \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \frac{\sum_i w_i (n_i - 2u_4 p_i)}{\sum_i w_i} \quad u_0 = \frac{-\sum_i w_i ([u_1 u_2 u_3] + p_i^T) \cdot p_i}{\sum_i w_i} \end{array} \right.$$

Surface de points algébrique

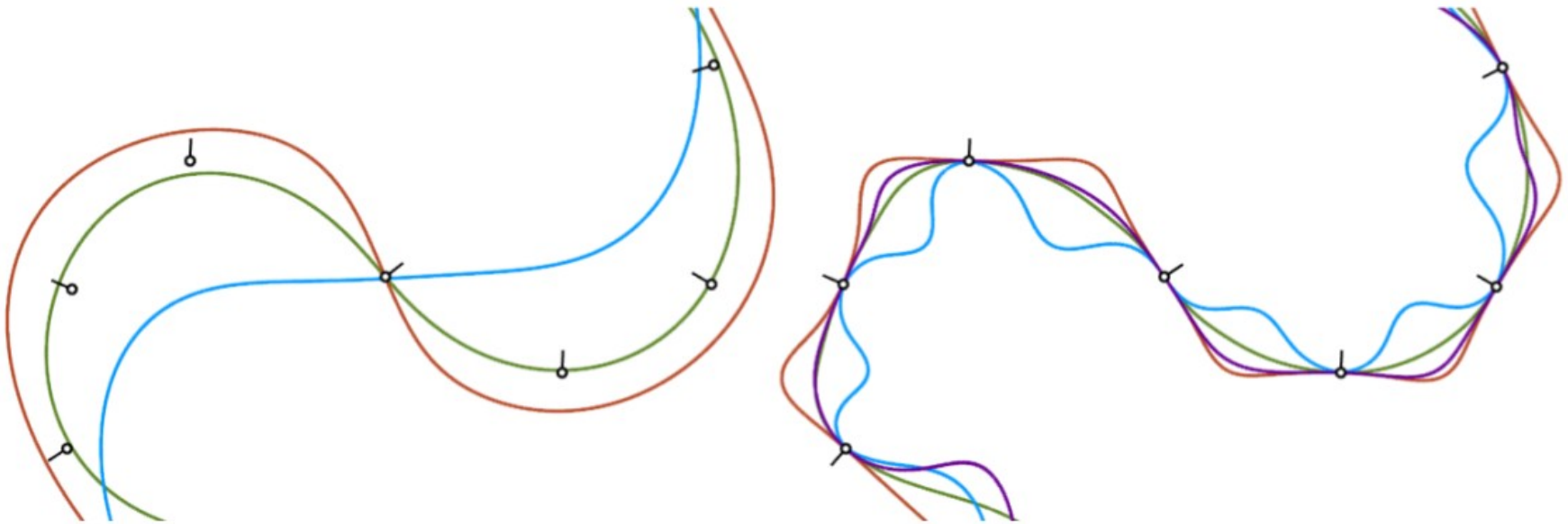
Note :

- Si on force $u_4 = 0$, on cherche le meilleur plan défini par les points les plus proches de x , avant de projeter x dessus.
- On retrouve dans ce cas la le modèle SPSS (qui est moins bon que HPSS...)
- → APSS étend le modèle SPSS (d'une autre manière que HPSS)

Surface de points algébrique



Surface de points algébrique



SPSS

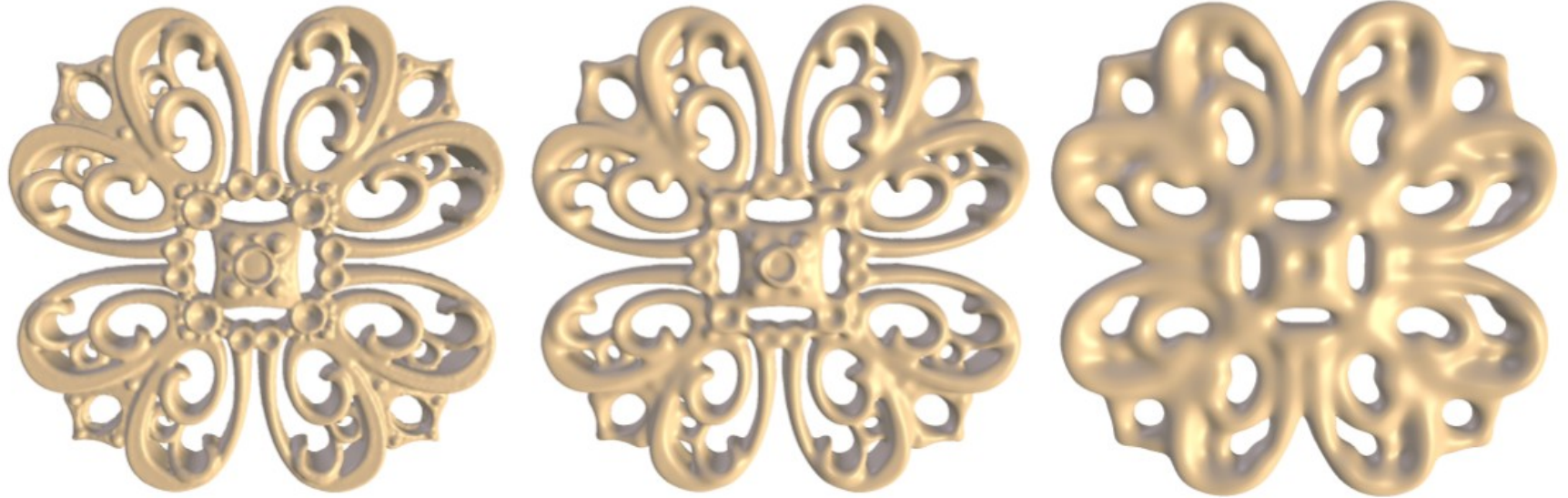
IMLS

HPSS

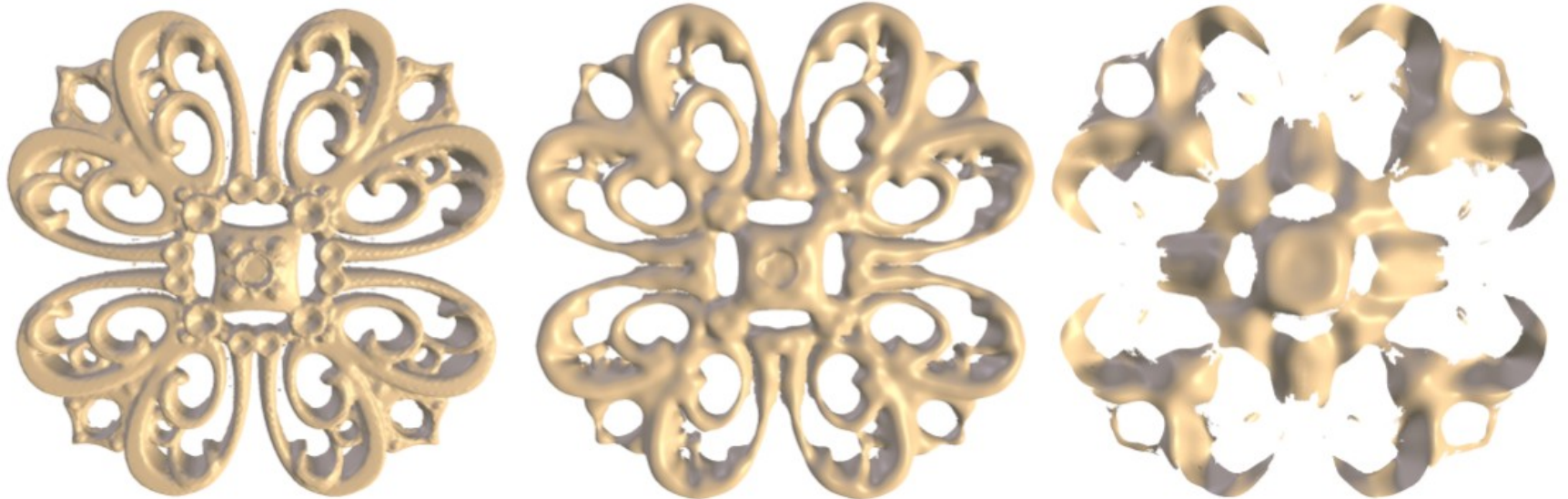
APSS

Taille du noyau approximant variant

APSS

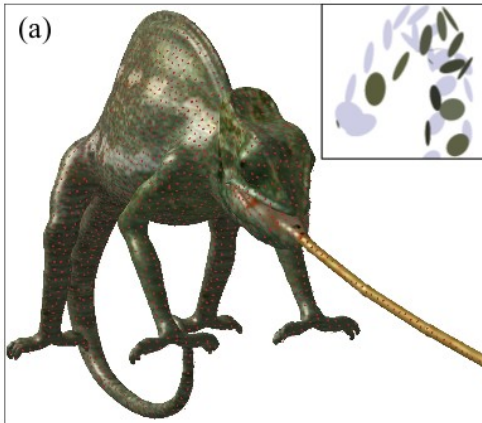


SPSS

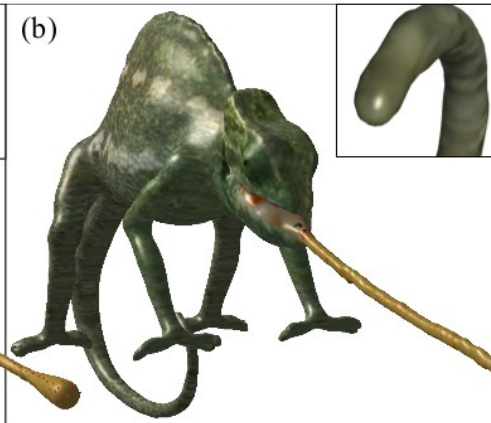


Comparison

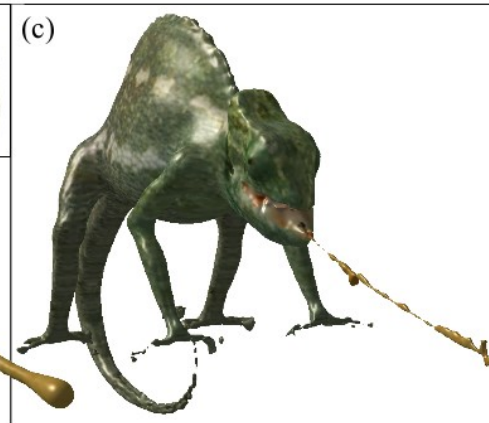
Downsampling



APSS



SPSS



IMLS



Édition de courbure

$$u_4 = \beta \frac{1}{2} \frac{(\sum_i w_i p_i^T \cdot n_i) - (\sum_i w_i p_i)^T \cdot (\sum_j w_j n_j) / (\sum_k w_k)}{(\sum_i w_i p_i^T \cdot p_i) - (\sum_i w_i p_i)^T \cdot (\sum_j w_j p_j) / (\sum_k w_k)}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \frac{\sum_i w_i (n_i - 2u_4 p_i)}{\sum_i w_i}$$

$$u_0 = \frac{-\sum_i w_i ([u_1 u_2 u_3] + p_i^T) \cdot p_i}{\sum_i w_i}$$



$\beta=8$

$\beta=1$

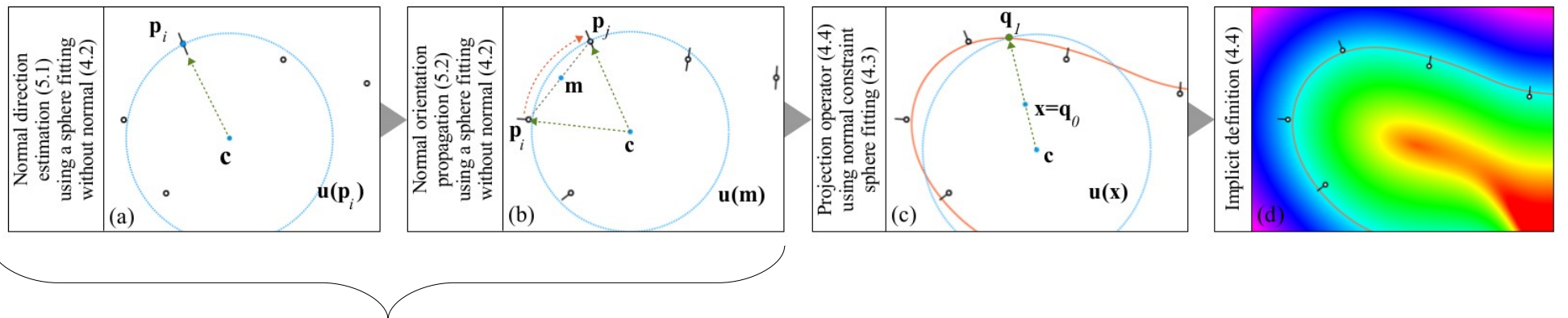


$\beta=1$

$\beta=-4.5$

Estimation des normales et fonction implicite

$$f(X) = (1, X^T, \|X\|^2) \cdot U(X)$$



Pour les nuages non-orientés, les auteurs proposent de fitter des sphères aux points sans les normales, de définir la normale ainsi, et de propager l'orientation (distinction des zones convexes et concaves)

Questions ?

