

Covariance Trees

Faster *Learning* & *Querying* for
Multi-Scale Local Gaussian Models

Thierry Guillemot **Andrés Almansa** Tamy Boubekeur



LTCI

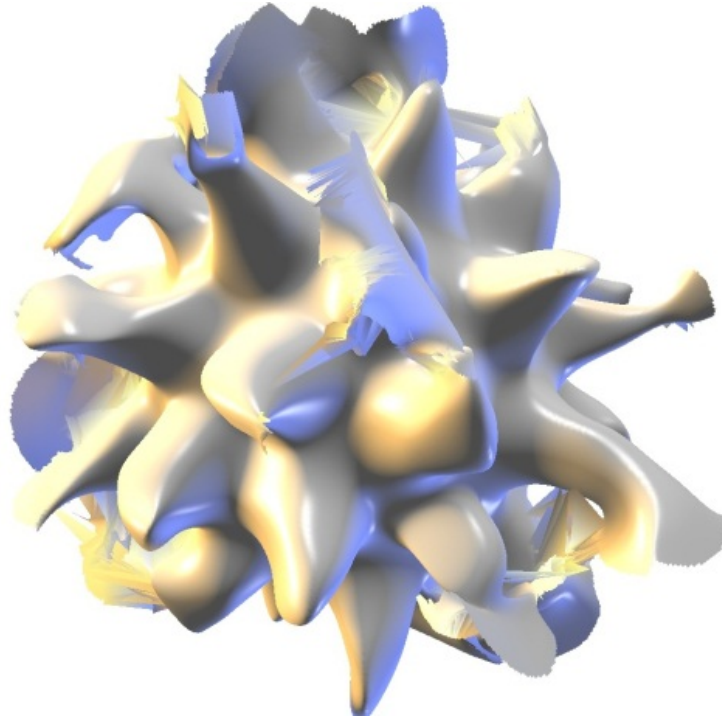


Journées Bordelaises d'Analyse Mathématique des Images

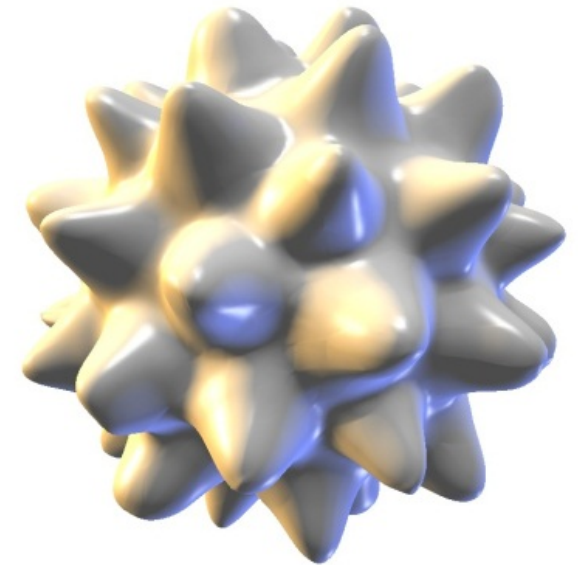
Surface reconstruction



3D point cloud



Point Set Surfaces
RIMLS [Öztireli et al. 2009]



Non-local Point Set Surface

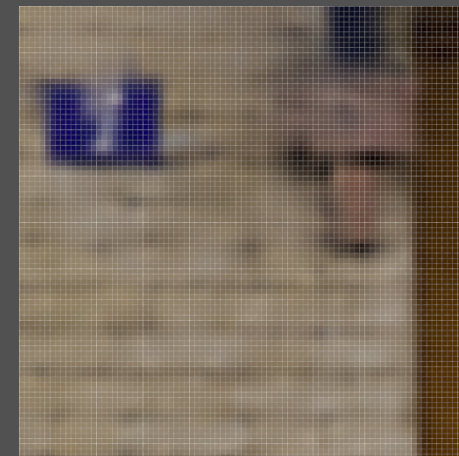
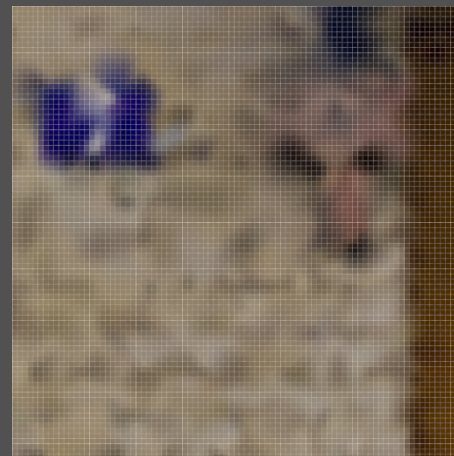
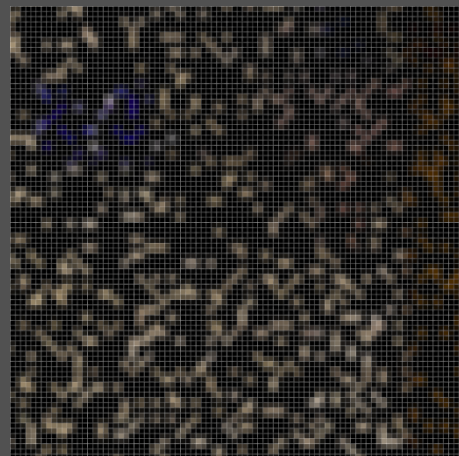
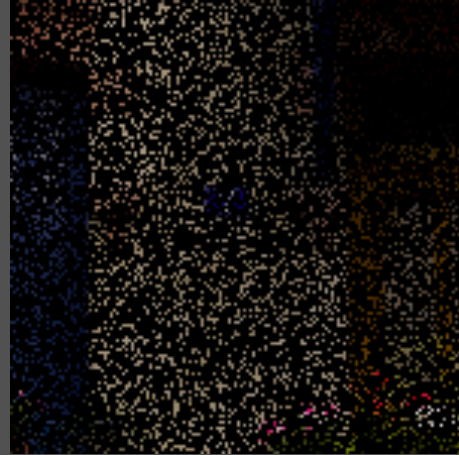
Reconstruction from random samples

Original

20% of original pixels

Cubic

CovTree



PSNR : 24.9 dB

PSNR : 27.2 dB

Outline

- Motivation
 - Example-based image inpainting
 - From example-based to model-based
 - Local Gaussian Models vs GMM
- Covariance Tree
- Applications
 - Learning-based Image denoising
 - Random Sampling
 - Surface reconstruction from 3D point clouds

Example-based image inpainting [Efros-Leung 1999, Wexler 2005]

Input: Visible part of the image $u|_{O^c}$

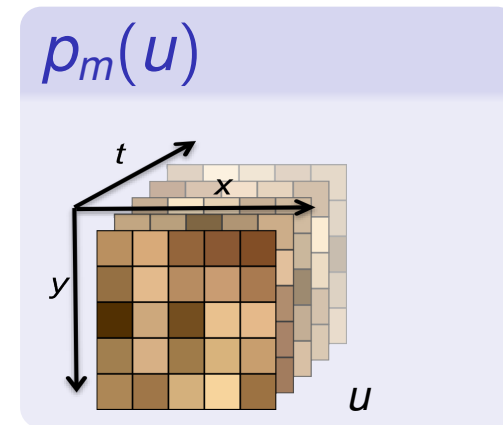
Output: reconstruction of the *occluded part* $u|_O$ via

$$\min_{u|_O} \sum_{m \in O} \|p_m(u) - p_{\varphi(m)}(u)\|^2$$

where

$$\varphi(m) = \arg \min_{n \in O^c} \|p_m(u) - p_n(u)\|^2$$

is the *nearest neighbour* of $p_m(u)$:
patches in Images and videos:



Example-based image inpainting [Arias-Caselles-Facciolo 2012]

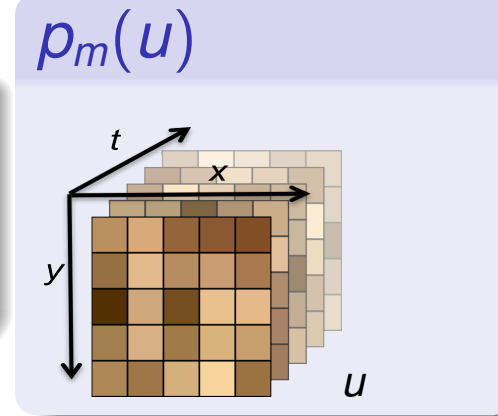
Input: Visible part of the image $u|_{O^c}$

Output: reconstruction of the *occluded part* $u|_O$ via

$$\min_{w, u|_O} \sum_{m \in O, n \in O^c} w(m, n) \|p_m(u) - p_n(u)\|^2 - T \sum_m H(w(m, \cdot))$$

under the constraint $\sum_n w(m, n) = 1, \forall m \in O$

where $H(f) = -\sum_n f(n) \log(f(n))$ is the entropy of the probability density distribution f .



Example-based image inpainting [Arias-Caselles-Facciolo 2012]

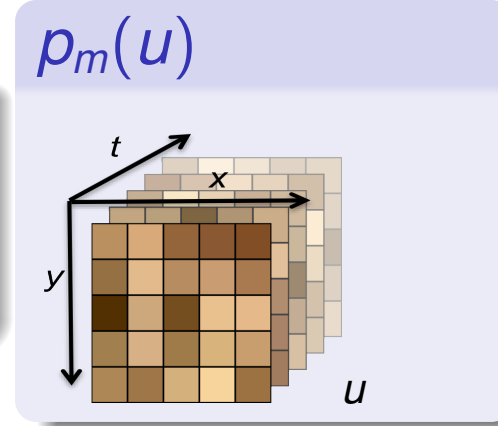
Input: Visible part of the image $u|_{O^c}$

Output: reconstruction of the occluded part $u|_O$ via

$$\min_{w, u|_O} \sum_{m \in O, n \in O^c} w(m, n) \|p_m(u) - p_n(u)\|^2 - T \sum_m H(w(m, \cdot))$$

under the constraint $\sum_n w(m, n) = 1, \forall m \in O$

where $H(f) = -\sum_n f(n) \log(f(n))$ is the entropy of the probability density distribution f .



Non-convex problem

Alternated minimisation of convex problems

- w -minimization (Learn local distribution)

$$w(m, n) = \frac{1}{Z} e^{-\frac{1}{T} \|p_m(u) - p_n(u)\|^2}$$

- u -min: (a posteriori expectation)

$$\hat{p}_m = E[p | p_m(u)] = \sum_n w(m, n) p_n$$

- Aggregation: $u(m) = \sum_n \hat{p}_n[n - m]$

Example-based image inpainting [Arias-Caselles-Facciolo 2012]

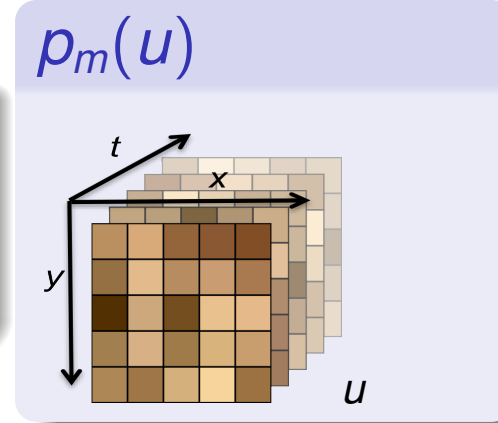
Input: Visible part of the image $u|_{O^c}$

Output: reconstruction of the occluded part $u|_O$ via

$$\min_{w, u|_O} \sum_{m \in O, n \in O^c} w(m, n) \|p_m(u) - p_n(u)\|^2 - T \sum_m H(w(m, \cdot))$$

under the constraint $\sum_n w(m, n) = 1, \forall m \in O$

where $H(f) = -\sum_n f(n) \log(f(n))$ is the entropy of the probability density distribution f .



Non-convex problem

Alternated minimisation of convex problems

- **w-minimization** (*Learn local distribution*)

$$w(m, n) = \frac{1}{Z} e^{-\frac{1}{T} \|p_m(u) - p_n(u)\|^2}$$

- **u-min:** (*a posteriori expectation*)

$$\hat{p}_m = E[p | p_m(u)] = \sum_n w(m, n) p_n$$

- **Aggregation:** $u(m) = \sum_n \hat{p}_n[n - m]$

Challenges

- Computation of w truncated and approximated by Patch Match [Barnes 2009] **Other alternatives? non-structured data?**
- Non-convexity: Multi-scale
- Patch similarity: ℓ^2 is ambiguous for fine textures

Example-based vs. model-based image inpainting

Example based

- w -minimization (*Learn local distribution*)

$$w(m, n) = \frac{1}{Z} e^{-\frac{1}{T} \|p_m(u) - p_n(u)\|^2}$$

- u -min: (*a posteriori expectation*)

$$\hat{p}_m = E [p \mid p_m(u)] = \sum_n w(m, n) p_n$$

- Aggregation: $u(m) = \sum_n \hat{p}_n[n - m]$

Model based

- w -minimization (*Learn local model*)

$w(m, \cdot) \sim N(\mu_m, \Sigma_m)$ that fits
 $\{p_n(u) : \|p_m(u) - p_n(u)\|^2 < T\}$

- u -minimization: estimate \hat{p}_n by:

- ▶ EAP (blurry), or...
- ▶ MAP, or...
- ▶ Random synthesis near $p_m(u)$ based on $N(\mu_m, \Sigma_m)$

- Aggregation: $u(m) = \sum_n \hat{p}_n[n - m]$

- fast algorithms on unstructured data (CovTree)

- synthesize vs. copy

Non-Local Means denoising [Buades-Coll-Morel 2005]

Input: Noisy image $\tilde{u} = u + n$ where $n \sim N(0, \sigma^2 Id)$.

Output: Estimated clean image \hat{u} via

$$\max_u \sum_{m, n} w(m, n) \|p_m(u) - p_n(\tilde{u})\|^2 - T \sum_m H(w(m, \cdot))$$

under the constraint $\sum_n w(m, n) = 1, \forall m \in O$

Example based

- w -minimization (*Learn local distribution*)

$$w(m, n) = \frac{1}{Z} e^{-\frac{\|p_m(u) - p_n(\tilde{u})\|^2 - T}{T}}$$

- u -minimization: (*a posteriori expectation*)

$$\hat{p}_m = \sum_n w(m, n) p_n$$

- Aggregation: $\hat{u}(m) = \sum_n \hat{p}_n[n - m]$

Non-Local Bayes denoising [Lebrun-Buades-Morel 2013]

Input: Noisy image $\tilde{u} = u + n$ where $n \sim N(0, \sigma^2 Id)$.

Output: Estimated clean image \hat{u} via

$$\max_{u(m), N(\mu_m, \Sigma_m)} \Pr [p_m(u) \mid p_n(\tilde{u}), N(\mu_m, \Sigma_m)]$$

Model based

- *w*-minimization (Learn Local Gaussian Model)

$$\mu_m = \frac{1}{Z} \sum_n e^{-\frac{\|p_m(u) - p_n(\tilde{u})\|^2}{\sigma^2}} p_n(\tilde{u})$$
$$\Sigma_m = \frac{1}{Z} \sum_n e^{-\frac{\|p_m(u) - p_n(\tilde{u})\|^2}{\sigma^2}} \bar{p}_n(\tilde{u}) \bar{p}_n(\tilde{u})^T - \sigma^2 Id$$

- *u*-minimization: (MAP)

$$\hat{p}_m = \arg \min_q \frac{1}{\sigma^2} \|q - p_m(\tilde{u})\|^2 + (q - \mu_m)^T \Sigma_m^{-1} (q - \mu_m)$$

- Aggregation: $\hat{u}(m) = \sum_n \hat{p}_n[n - m]$

Piecewise Linear Estimators [Yu-Mallat-Sapiro 2012]

Input: *Perturbed image* $\tilde{u} = Au + n$ where $n \sim N(0, \sigma^2 Id)$.

Output: *Restored image* \hat{u} via

$$\max_{u(m), N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), k(m)} \Pr [p_m(u) \mid p_n(\tilde{u}), N(\boldsymbol{\mu}_{k(m)}, \boldsymbol{\Sigma}_{k(m)})]$$

with $k = 1, \dots, 20$

Model based

- initialization: $\hat{u}^0, (\boldsymbol{\mu}_k^0, \boldsymbol{\Sigma}_k^0), k(m)$
- Relearn Gaussian Models $(\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i)$ to fit $\{p_m(u) : k(m) = k\}$
- Signal estimation (\hat{p}_m) and model selection ($k(m)$)

$$(\hat{p}_m, k(m)) = \arg \max_{q, k} \Pr [q \mid p_m(\tilde{u}), N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

- Aggregation: $\hat{u}^i(m) = \sum_n \hat{p}_n[n - m]$

Learning-based restoration [Zoran-Weiss 2011]

Offline learning

Input: a huge database of *natural image patches* $\mathbf{p}_i \in \mathcal{P}$.

Output: Gaussian Mixture Model $\{N(\mu_k, \Sigma_k) : k = 1, \dots, 250\}$ fitting the data
(several days worth of computation)

Restoration

Input: *Perturbed image* $\tilde{u} = Au + n$ where $n \sim N(0, \sigma^2 Id)$.

Gaussian Mixture Model $\{N(\mu_k, \Sigma_k) : k = 1, \dots, 250\}$
(representing the manifold of natural image patches)

Output: *Restored image* \hat{u} via

$$\max_{u(m), k(m)} \Pr [p_m(u) \mid p_n(\tilde{u}), N(\mu_{k(m)}, \Sigma_{k(m)})]$$

Covariance Tree [Guillemot-Almansa-Boubekeur 2014]

Learning

Input: a huge database of *data points* $\mathbf{p}_i \in \mathcal{P}$.

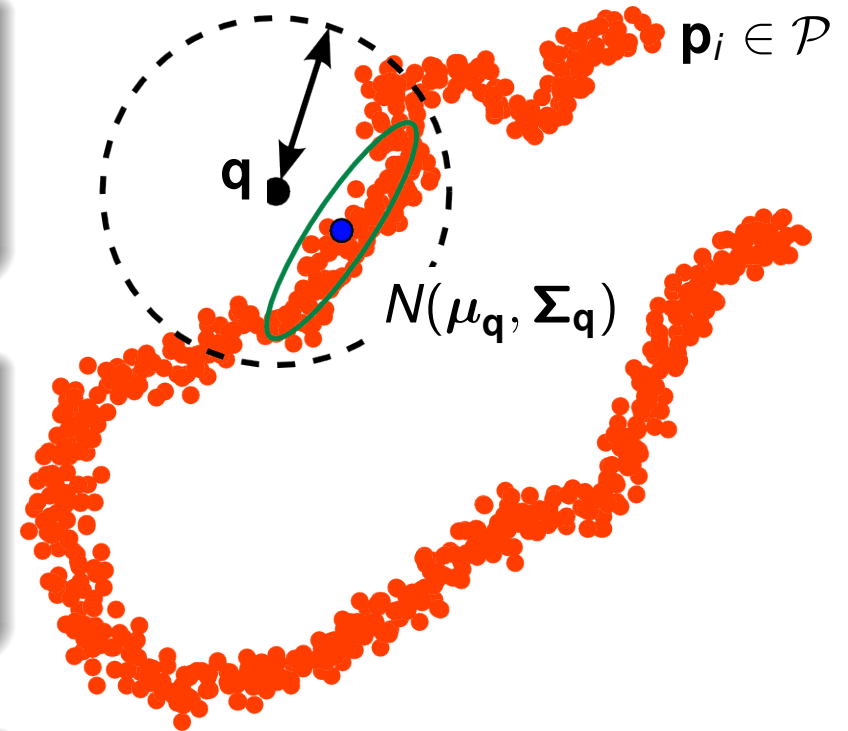
Output: pre-computed Local Gaussian Models at several *scales* and *locations*

Query

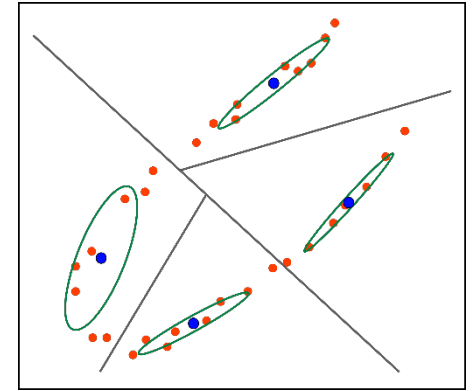
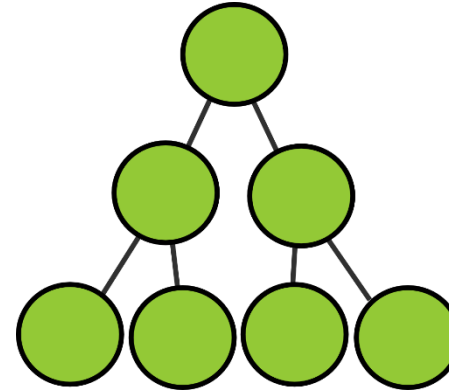
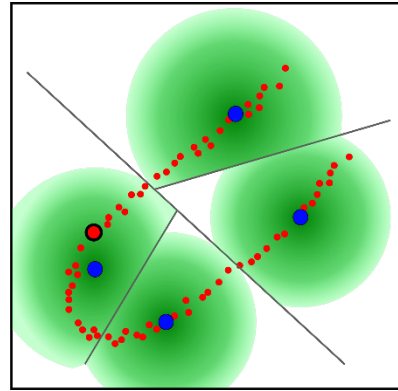
Input: a query point \mathbf{q} and a scale σ

Output: accurate approximation of $N(\mu_{\mathbf{q}}, \Sigma_{\mathbf{q}})$ fitting $\mathcal{P}|_{B(\mathbf{q}, \sigma)}$

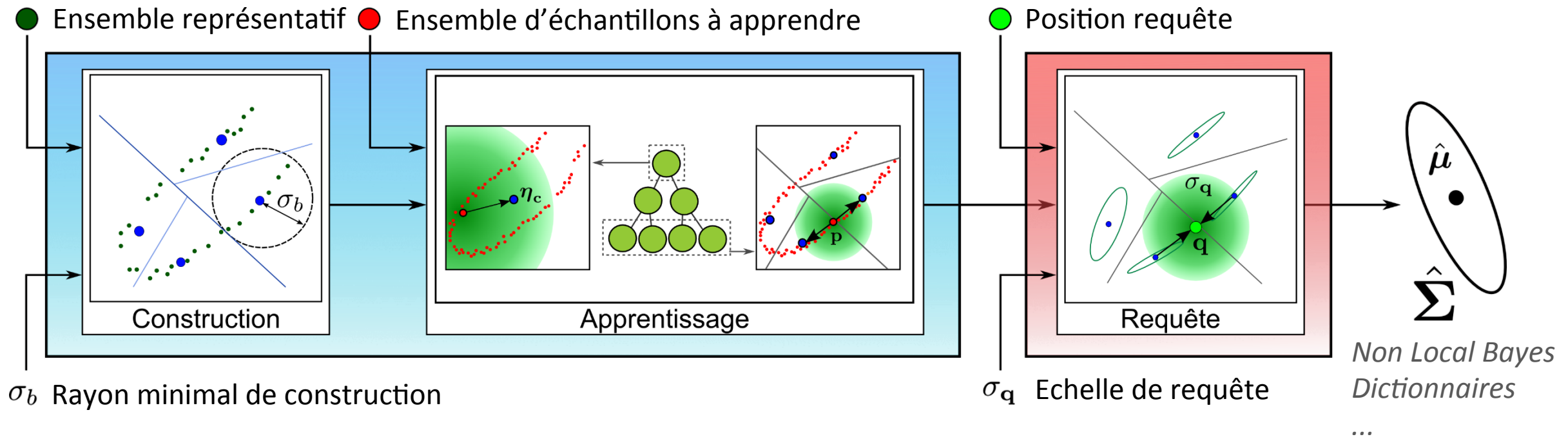
Bayesian Restoration



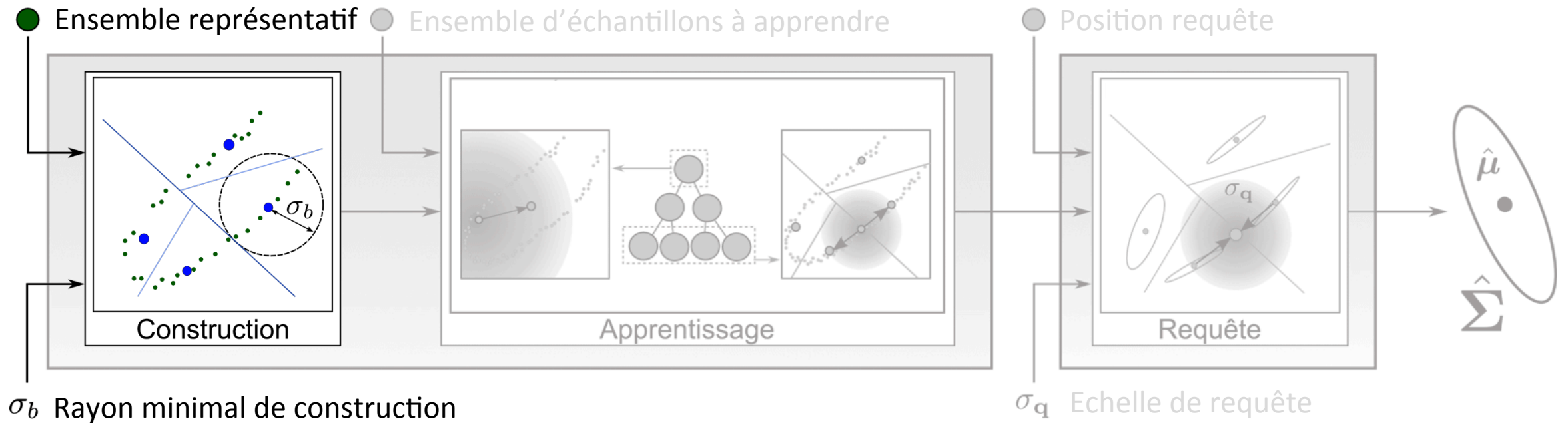
Covariance Tree



Outline of the algorithm

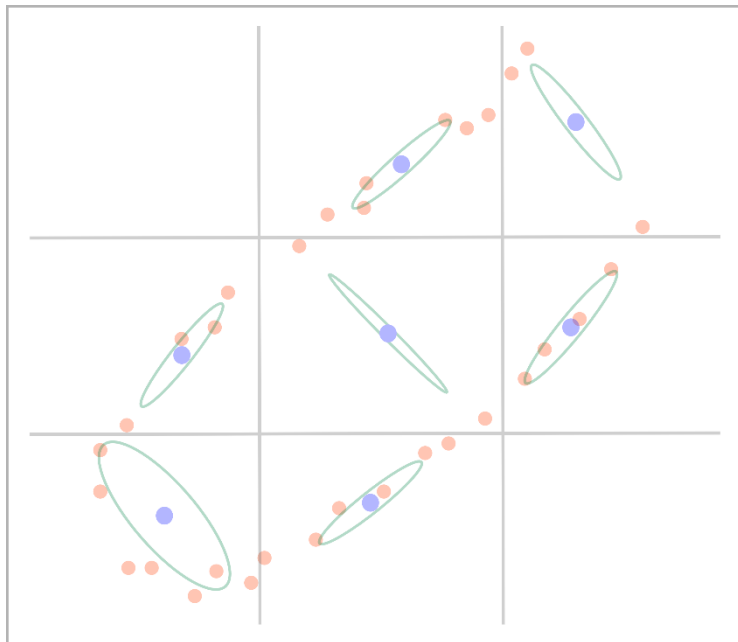


Outline of the algorithm

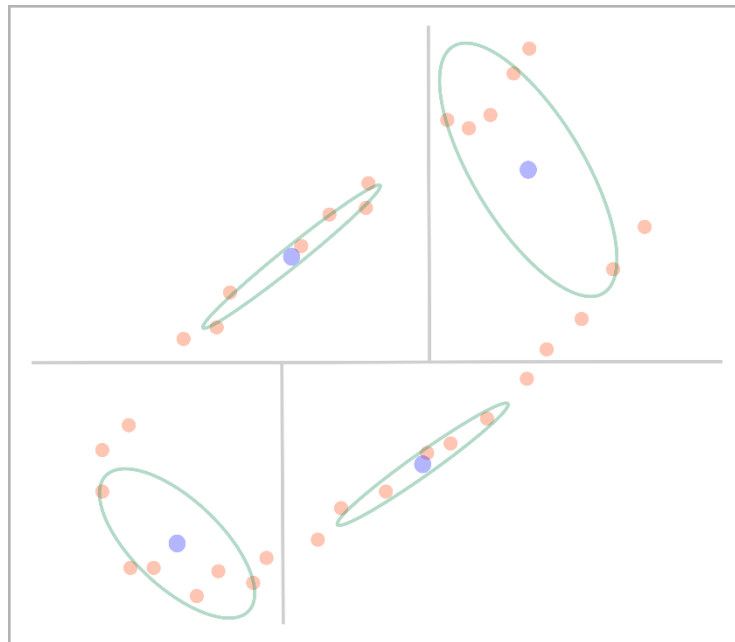


Partitioning into representative cells

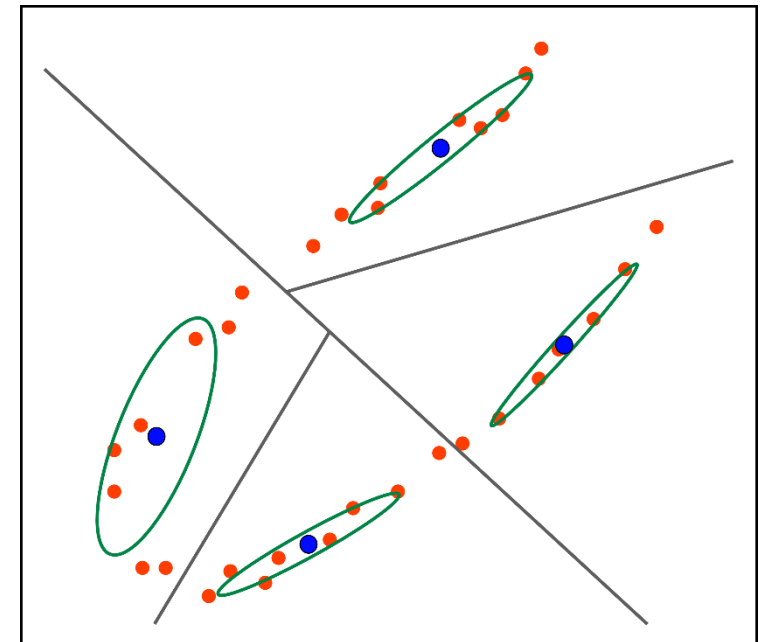
(Gaussian kd-trees [Adams et. al 2009])



Grille

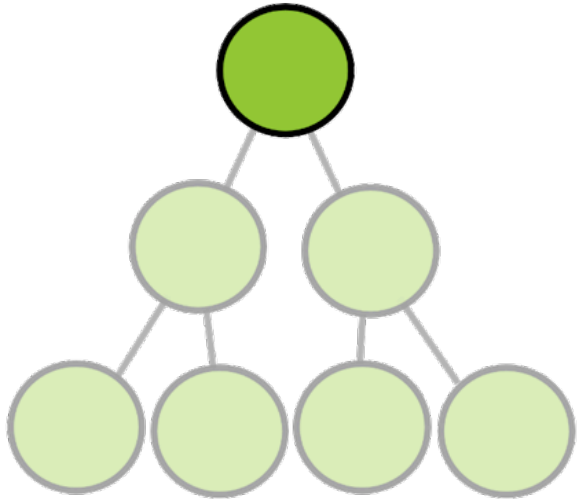


Kd-tree

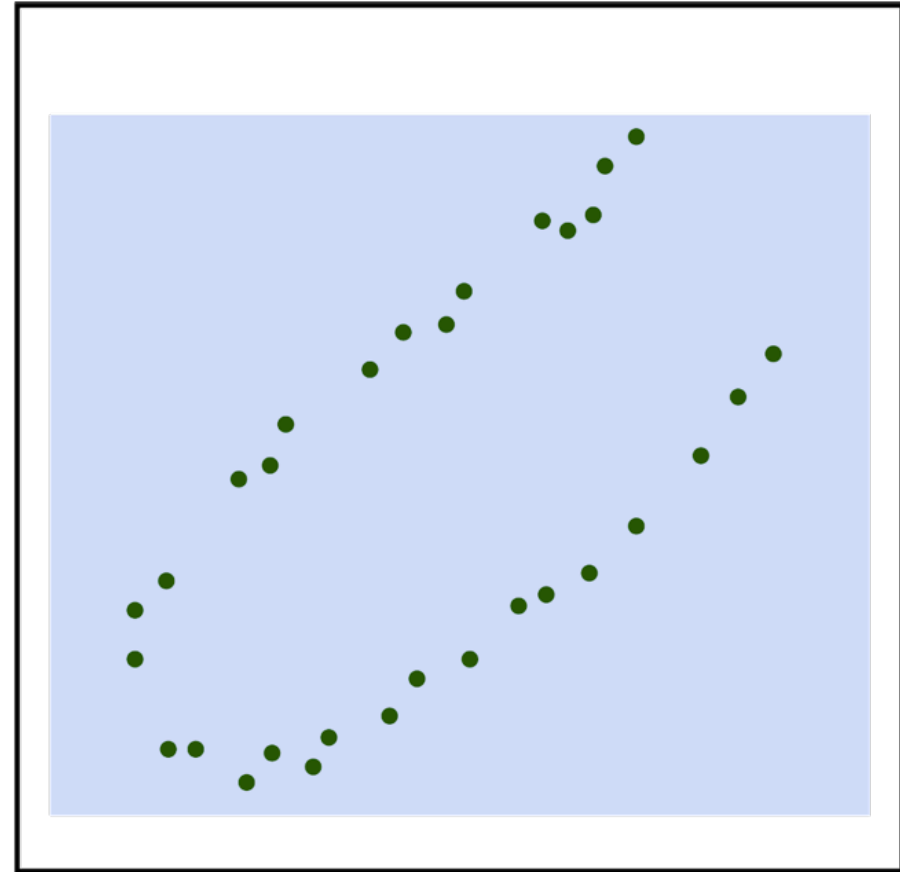


Bsp-tree

Building



Bsp-Tree



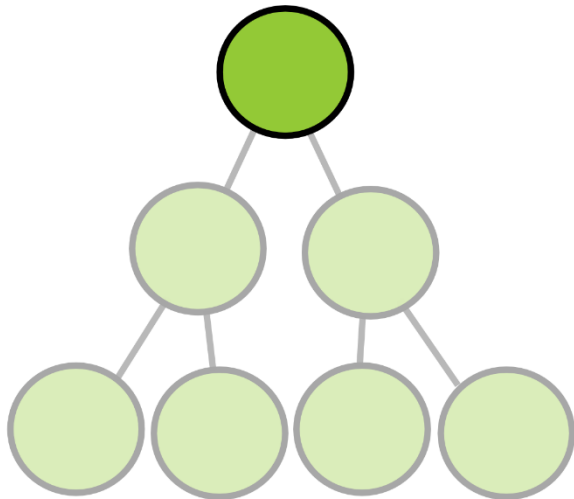
Echantillons

Building

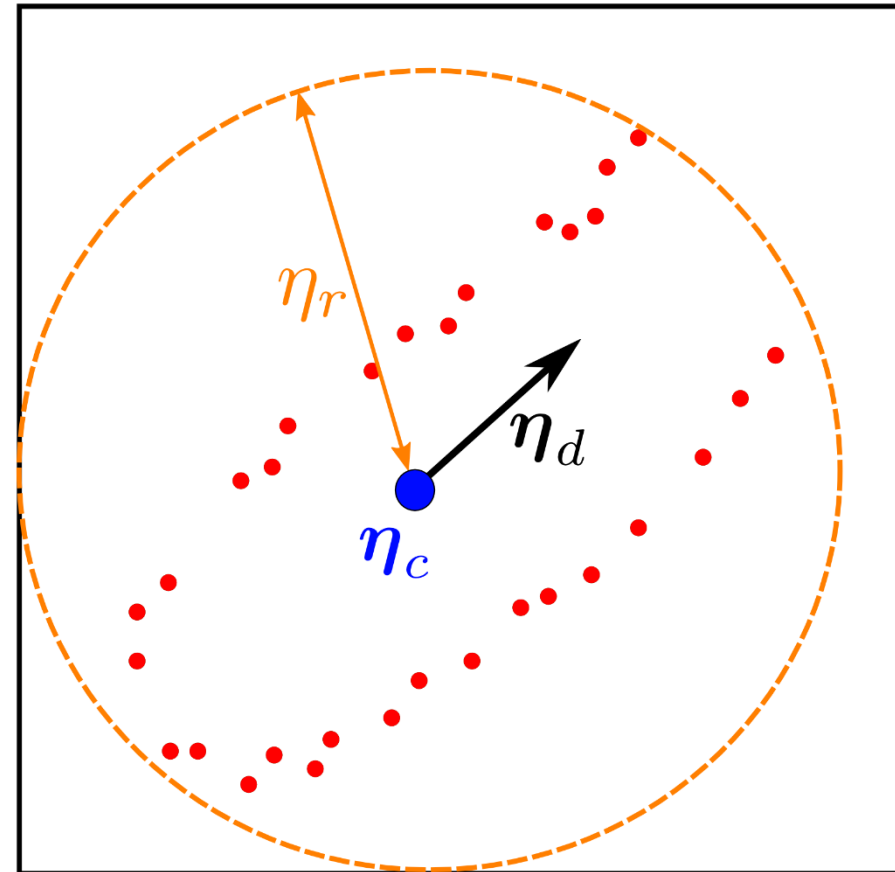
$$\eta_c = \text{moyenne}(\mathbf{p}_i)$$

$$\eta_r = \max_i \|\mathbf{p}_i - \eta_c\|$$

η_d = dir. plus gde variation des \mathbf{p}_i



Bsp-Tree



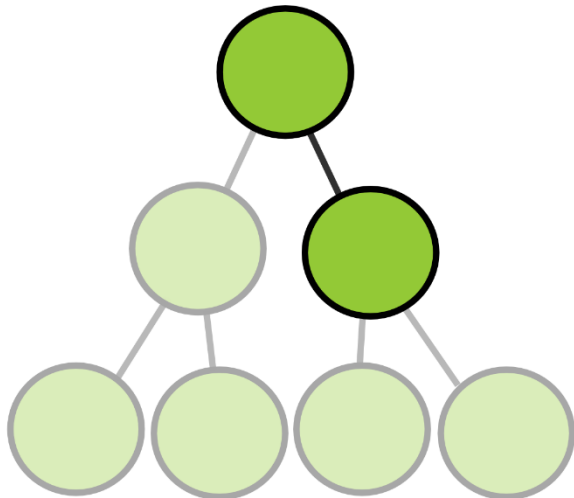
Echantillons

Building

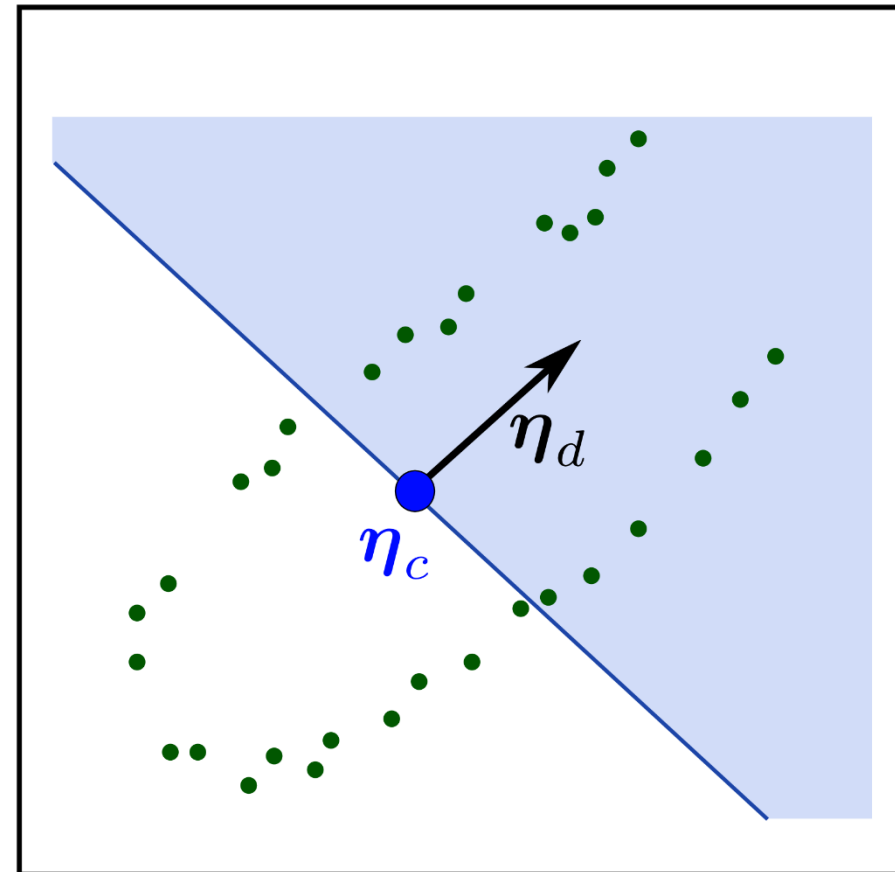
$$\eta_c = \text{moyenne}(\mathbf{p}_i)$$

$$\eta_r = \max_i \|\mathbf{p}_i - \eta_c\|$$

η_d = dir. plus gde variation des \mathbf{p}_i



Bsp-Tree



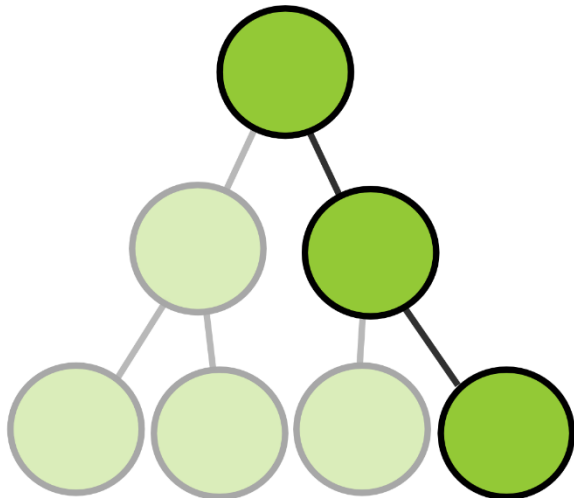
Echantillons

Building

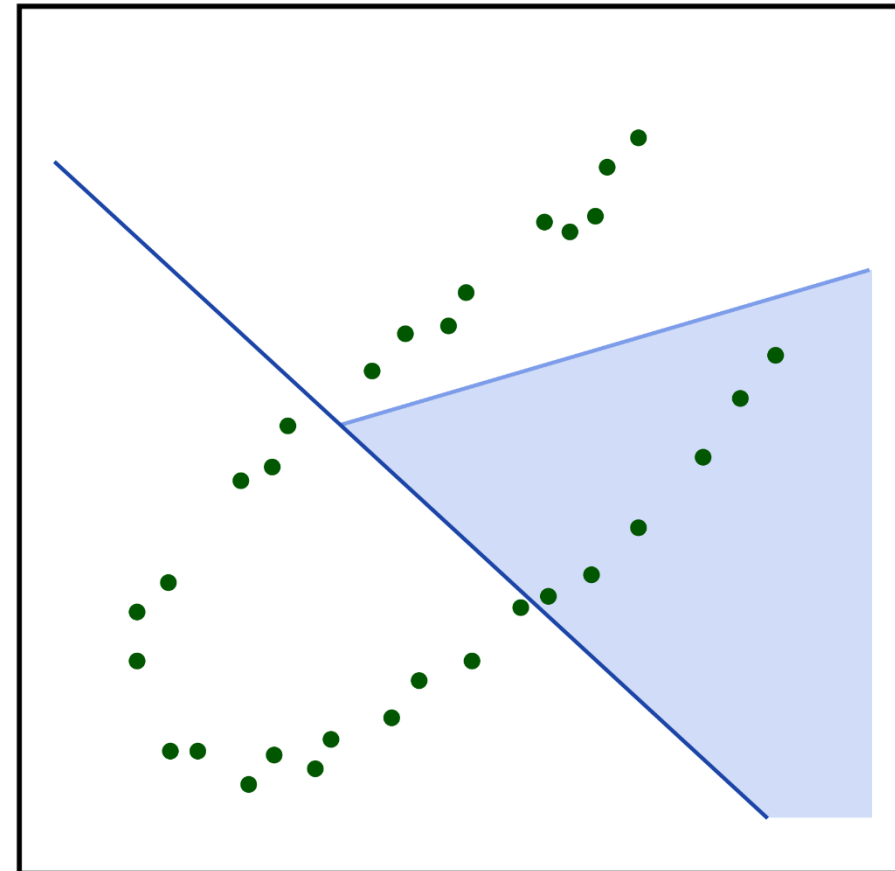
$$\eta_c = \text{moyenne}(\mathbf{p}_i)$$

$$\eta_r = \max_i \|\mathbf{p}_i - \eta_c\|$$

η_d = dir. plus gde variation des \mathbf{p}_i



Bsp-Tree



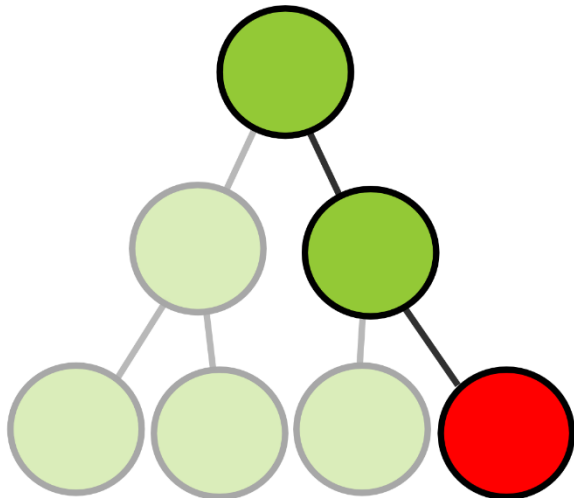
Echantillons

Building

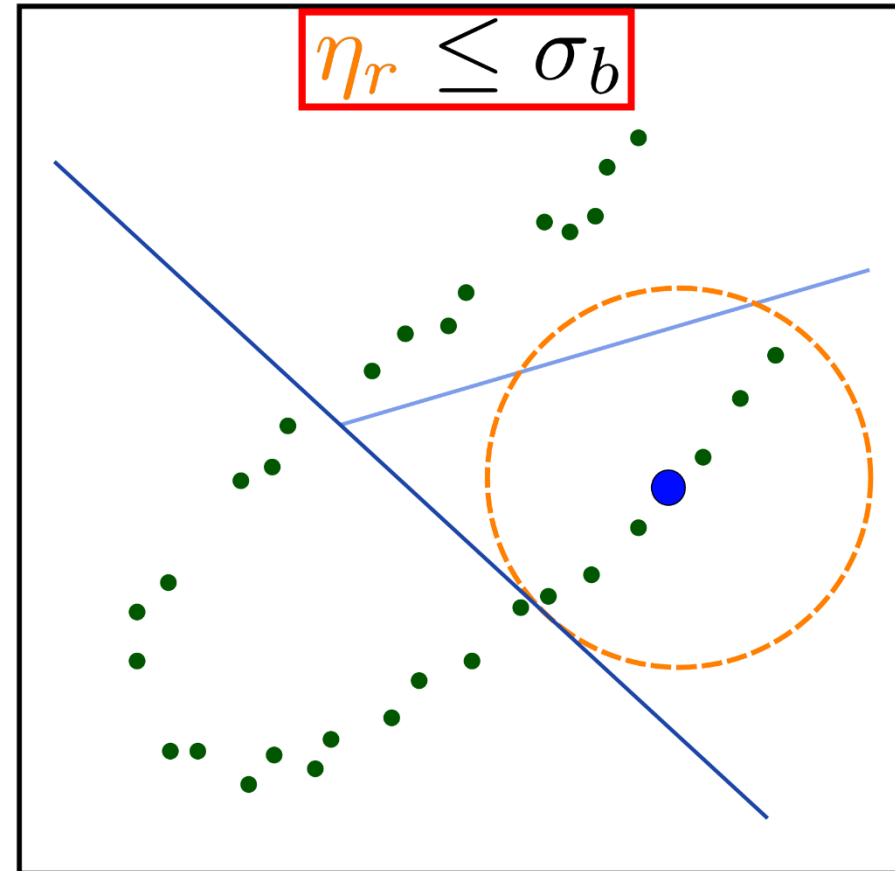
$$\eta_c = \text{moyenne}(\mathbf{p}_i)$$

$$\eta_r = \max_i \|\mathbf{p}_i - \eta_c\|$$

η_d = dir. plus gde variation des \mathbf{p}_i



Bsp-Tree



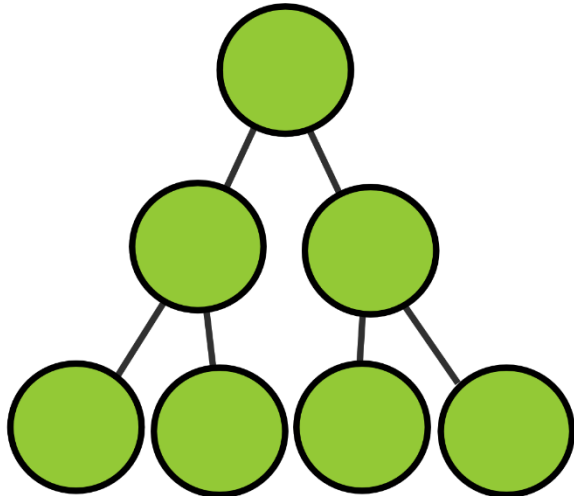
Echantillons

Building

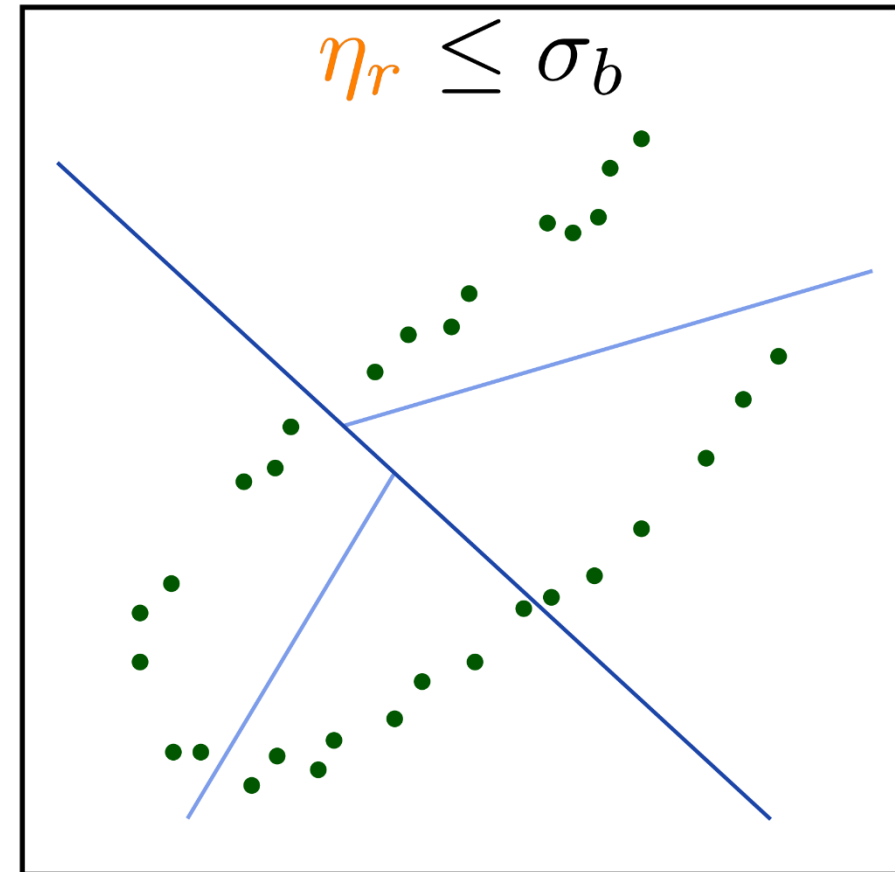
$$\eta_c = \text{moyenne}(\mathbf{p}_i)$$

$$\eta_r = \max_i \|\mathbf{p}_i - \eta_c\|$$

η_d = dir. plus gde variation des \mathbf{p}_i

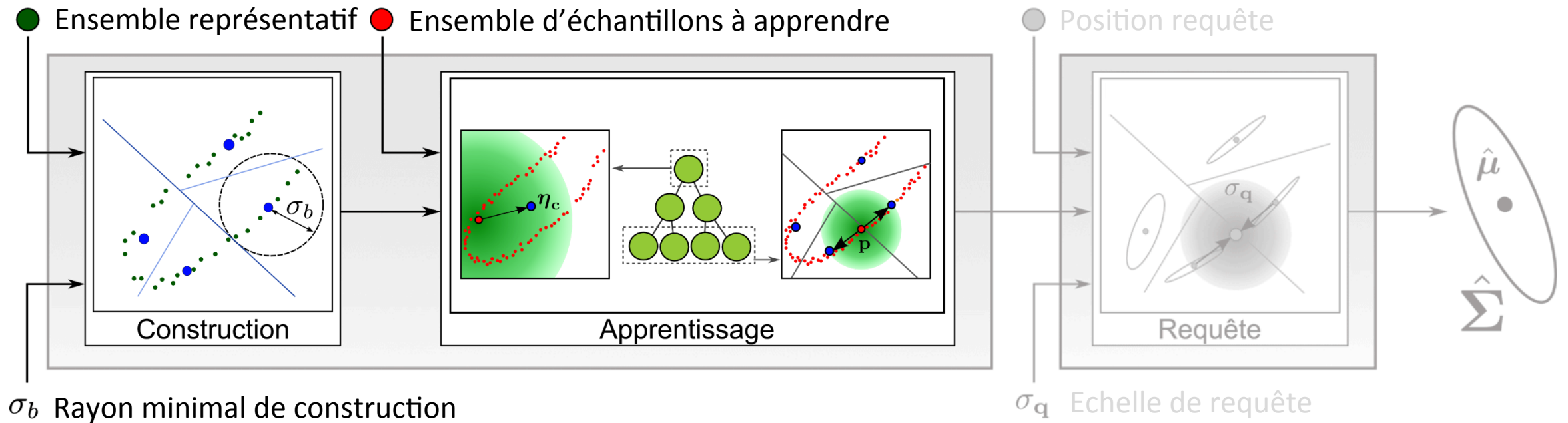


Bsp-Tree

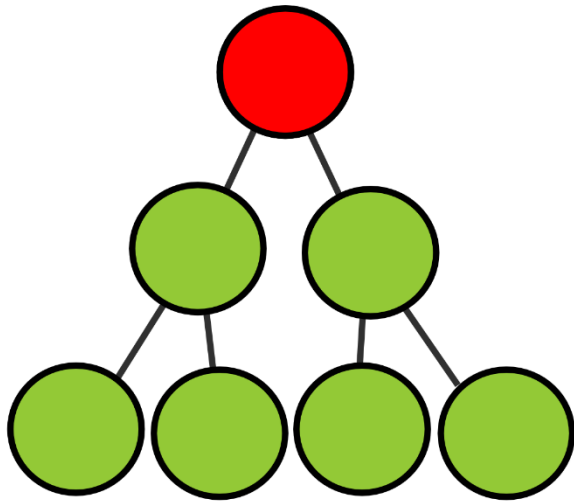


Echantillons

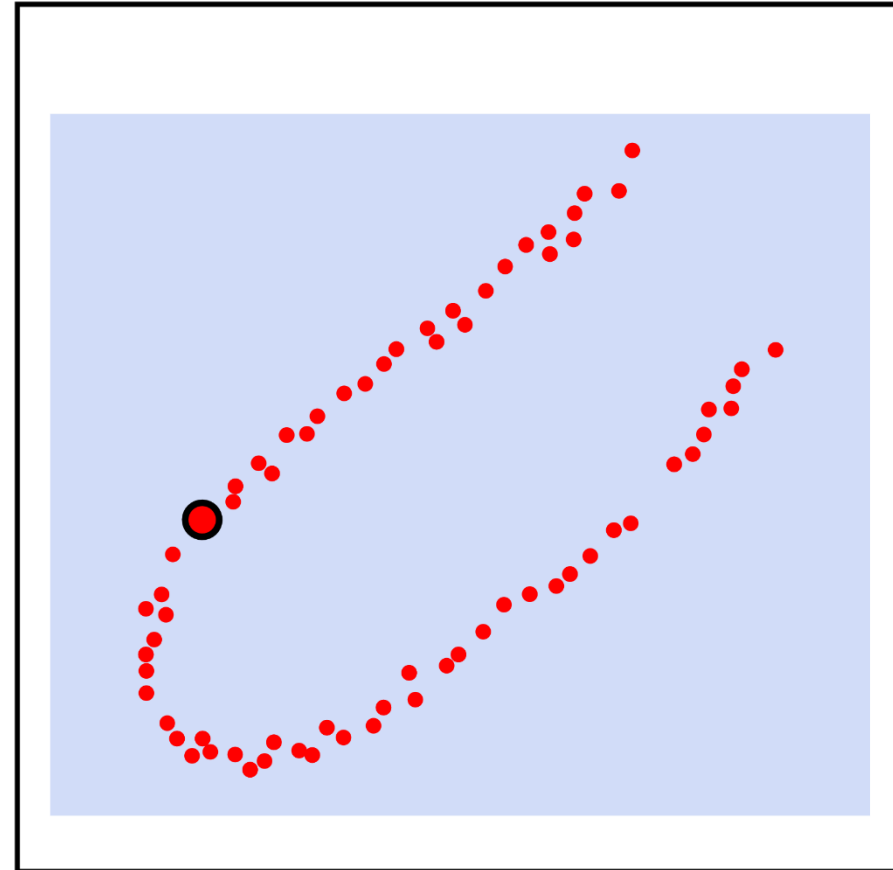
Outline of the algorithm



Learning



Bsp-Tree

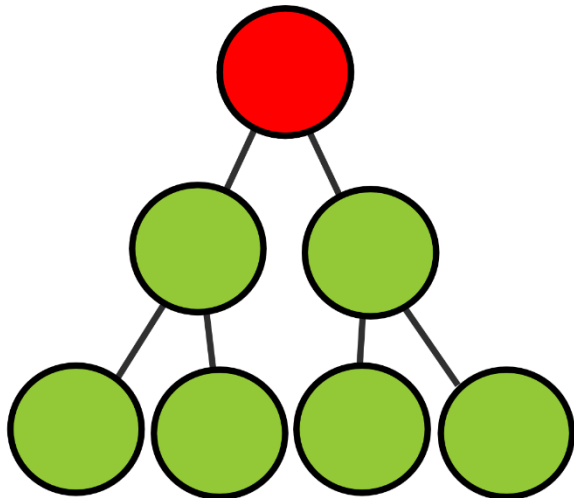


Echantillons à apprendre

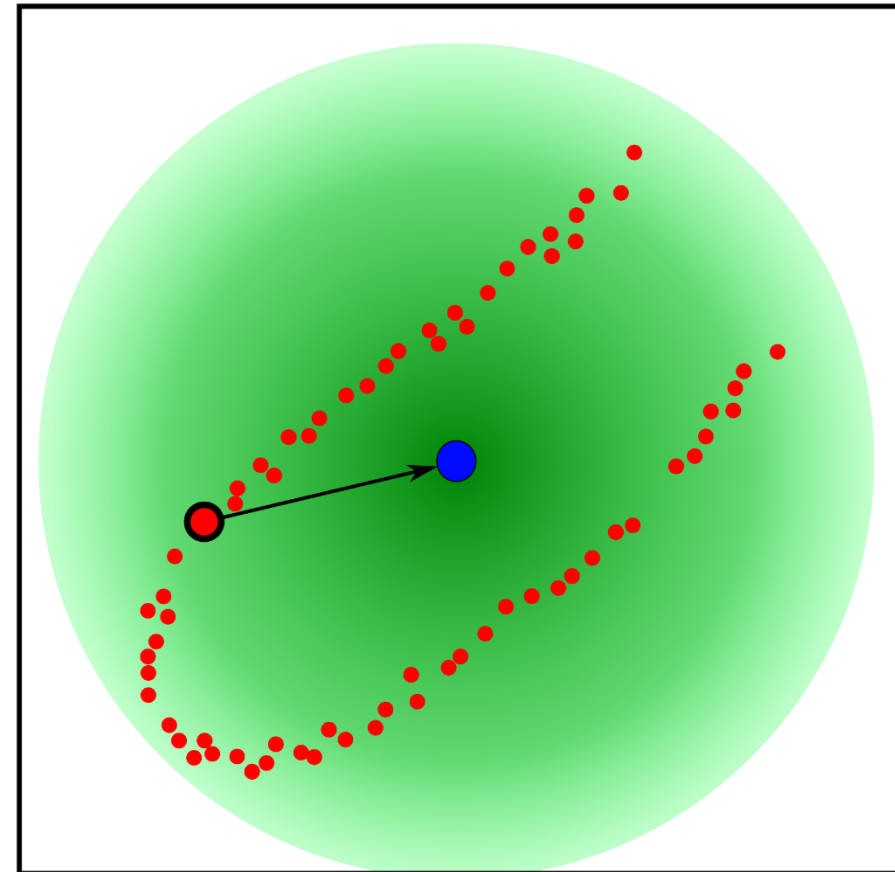
Learning

$$\boldsymbol{\mu}_\eta := \boldsymbol{\mu}_\eta + w_i \mathbf{p}_i$$

$$\boldsymbol{\Sigma}_\eta := \boldsymbol{\Sigma}_\eta + w_i \mathbf{p}_i^t \mathbf{p}_i$$



Bsp-Tree

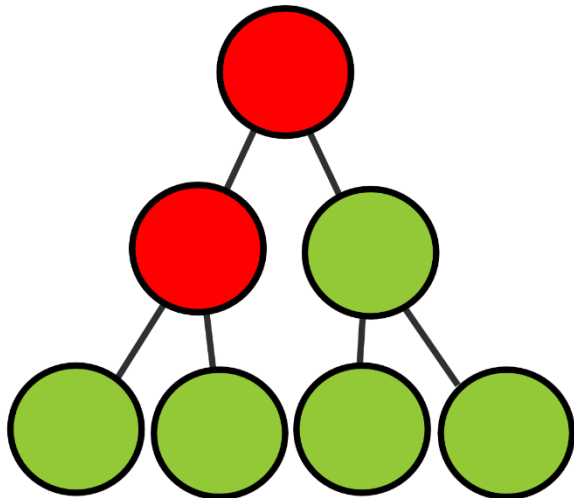


Echantillons à apprendre

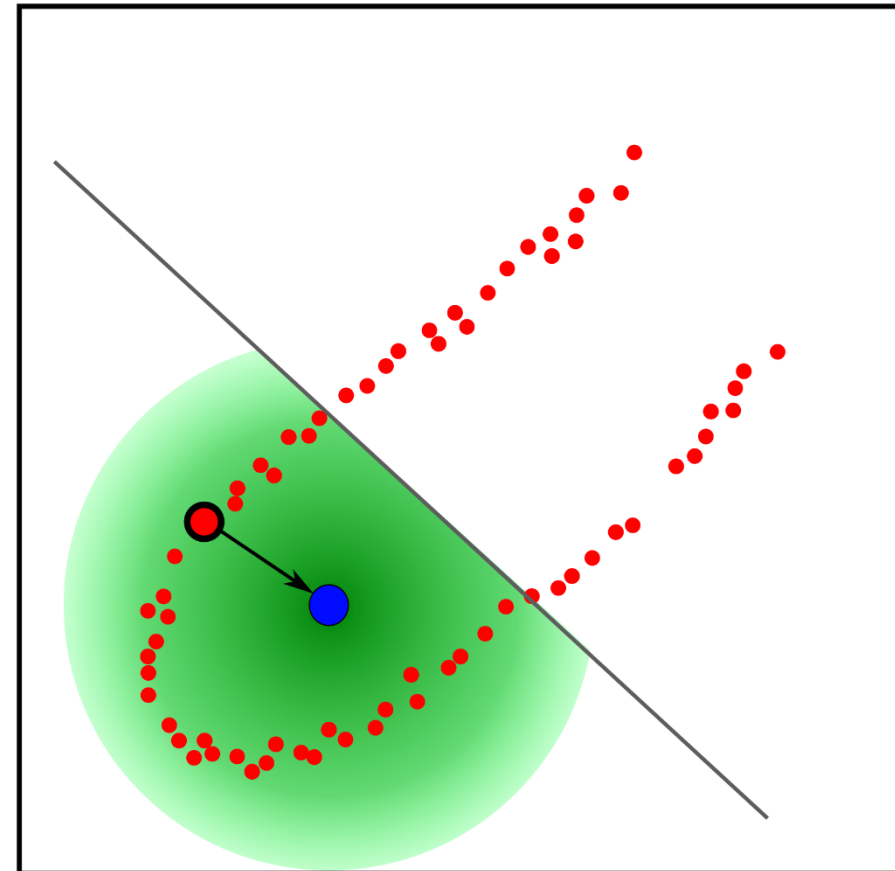
Learning

$$\mu_{\eta} := \mu_{\eta} + w_i \mathbf{p}_i$$

$$\Sigma_{\eta} := \Sigma_{\eta} + w_i \mathbf{p}_i^t \mathbf{p}_i$$



Bsp-Tree

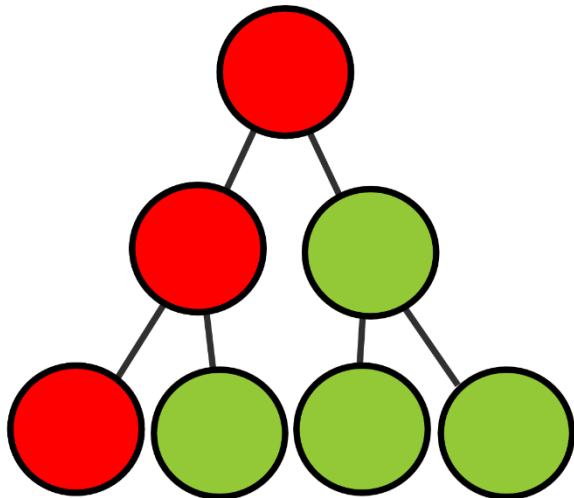


Echantillons à apprendre

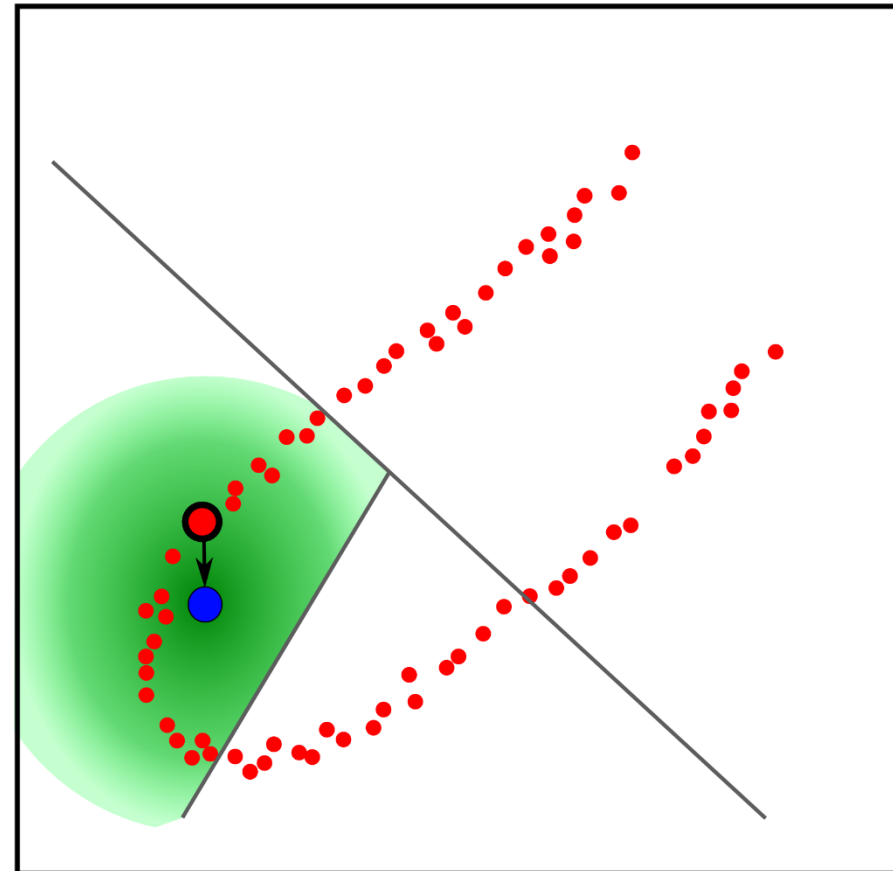
Learning

$$\mu_\eta := \mu_\eta + w_i \mathbf{p}_i$$

$$\Sigma_\eta := \Sigma_\eta + w_i \mathbf{p}_i^t \mathbf{p}_i$$



Bsp-Tree

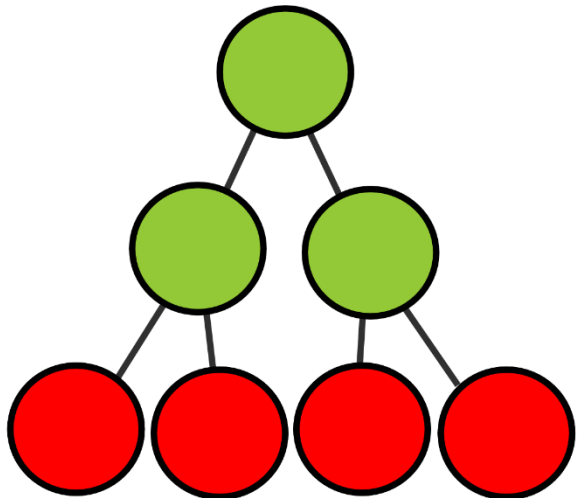


Echantillons à apprendre

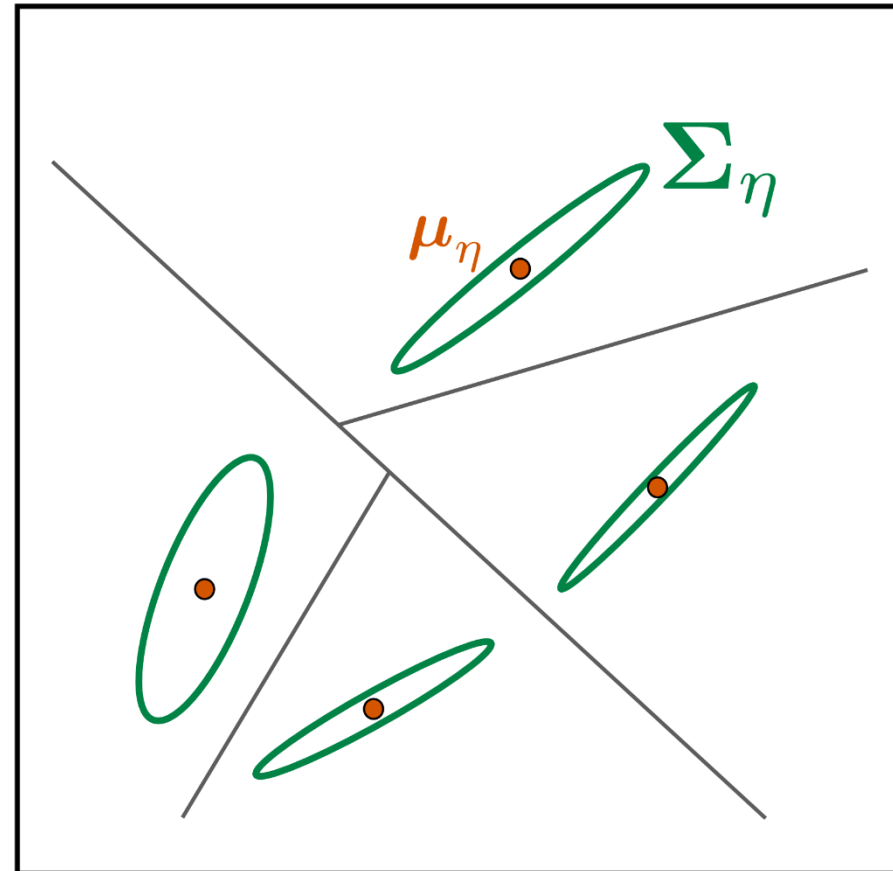
Apprentissage

$$\mu_\eta := \mu_\eta + w_i \mathbf{p}_i$$

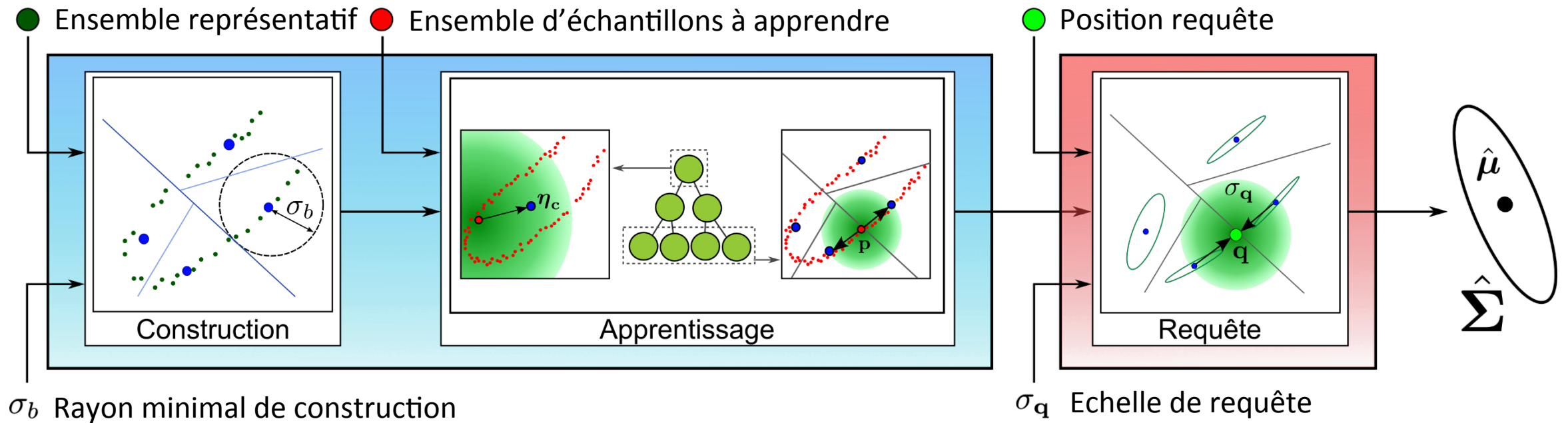
$$\Sigma_\eta := \Sigma_\eta + w_i \mathbf{p}_i^t \mathbf{p}_i$$



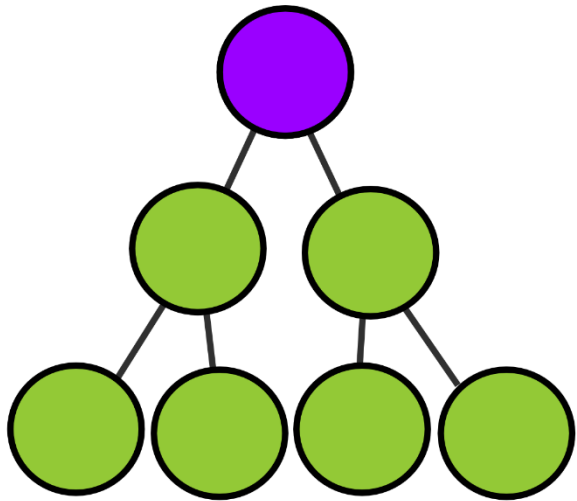
Bsp-Tree



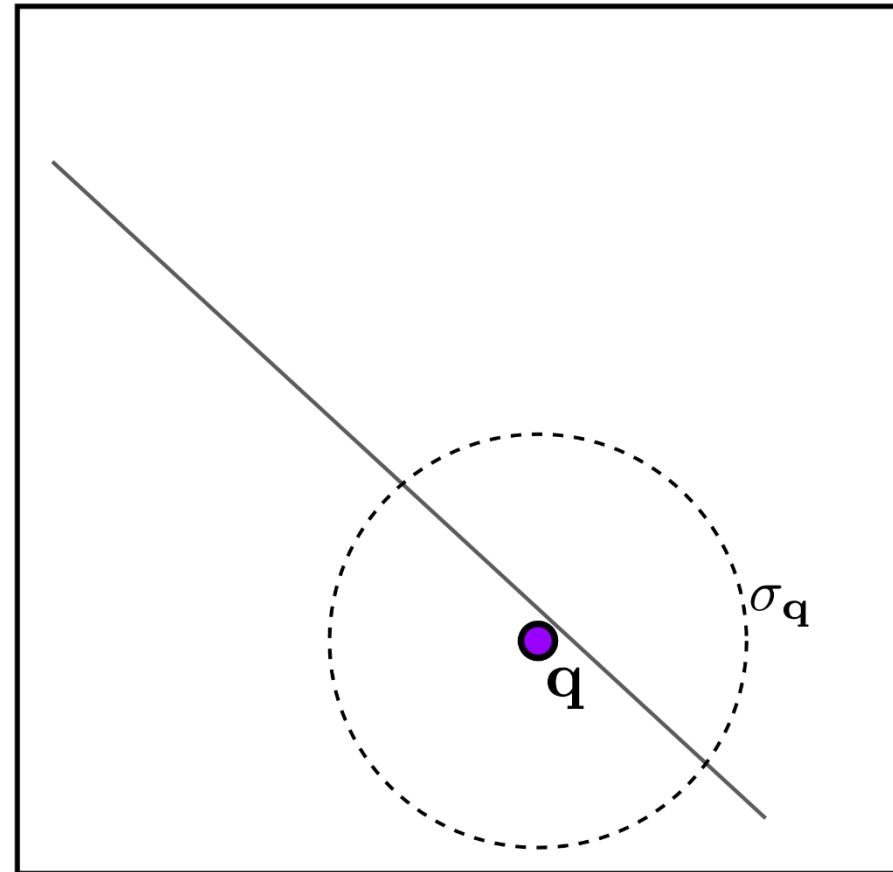
Outline of the algorithm



Query

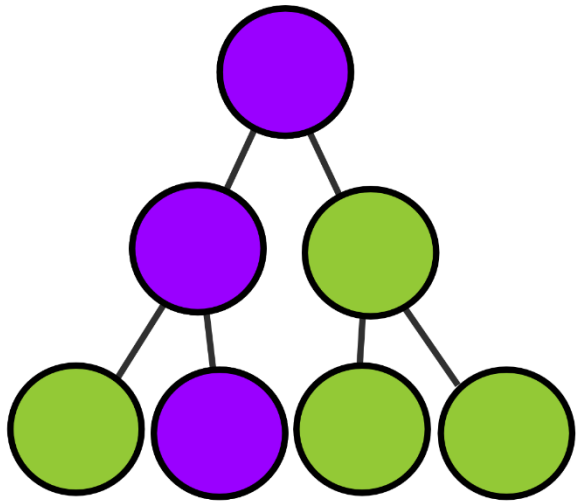


Bsp-Tree

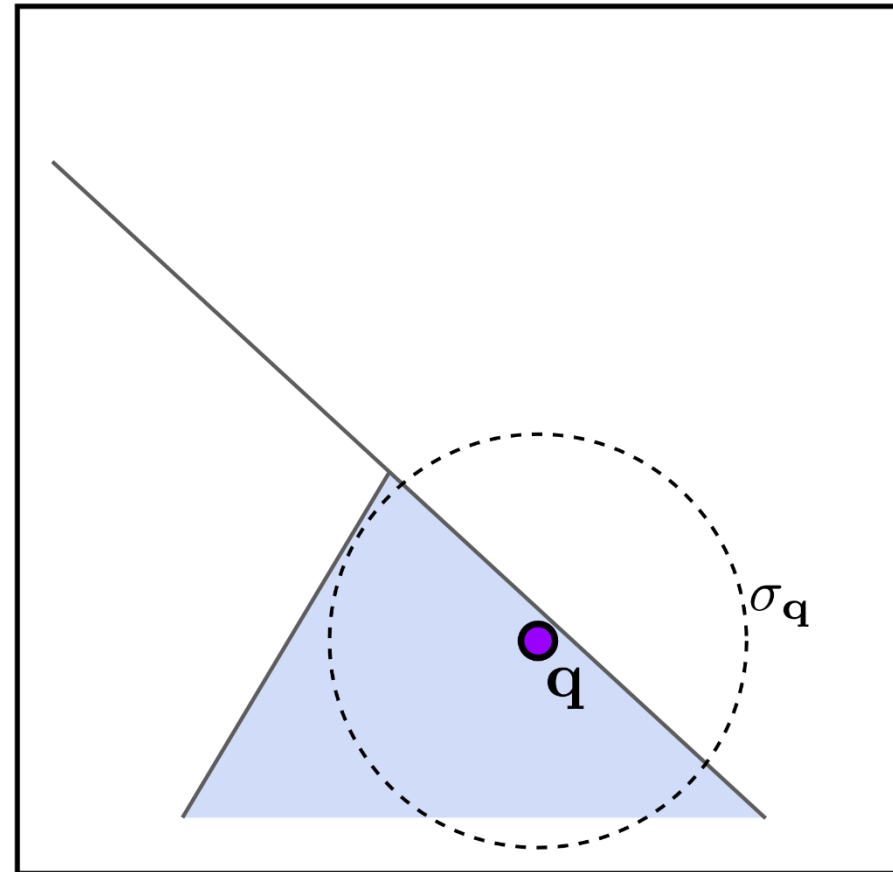


Requête

Query

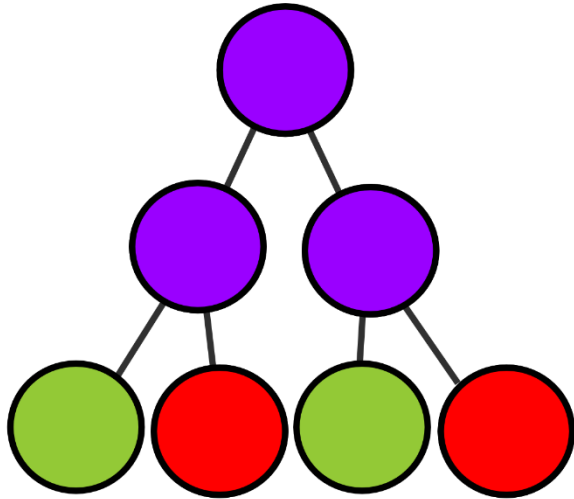


Bsp-Tree

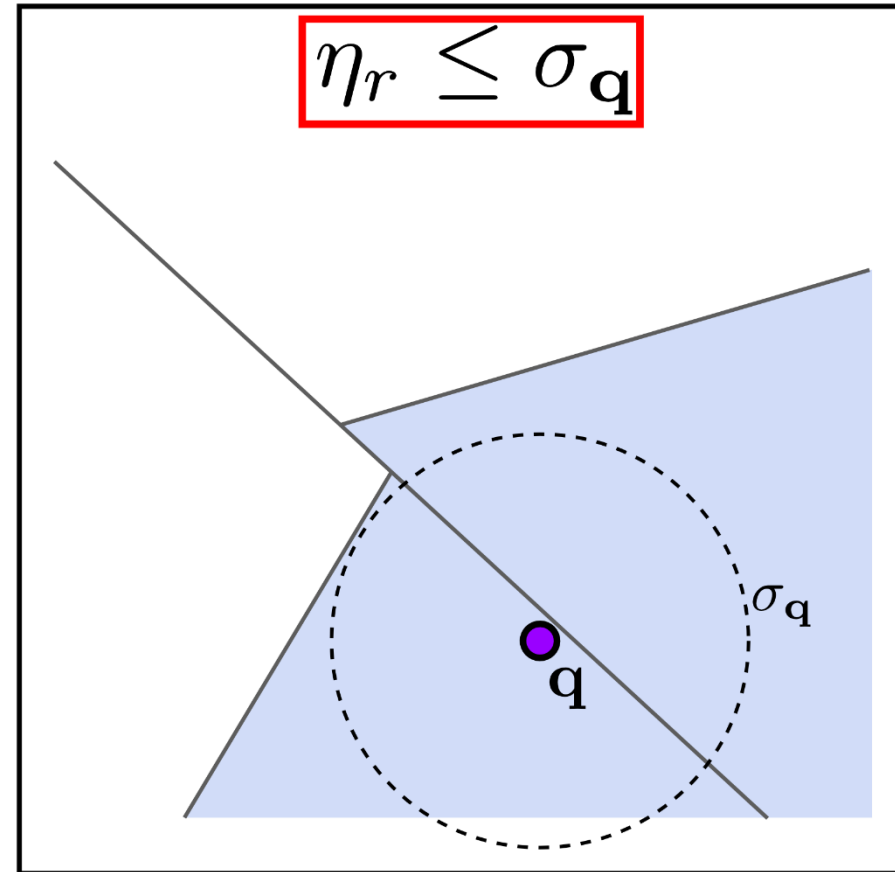


Requête

Query



Bsp-Tree

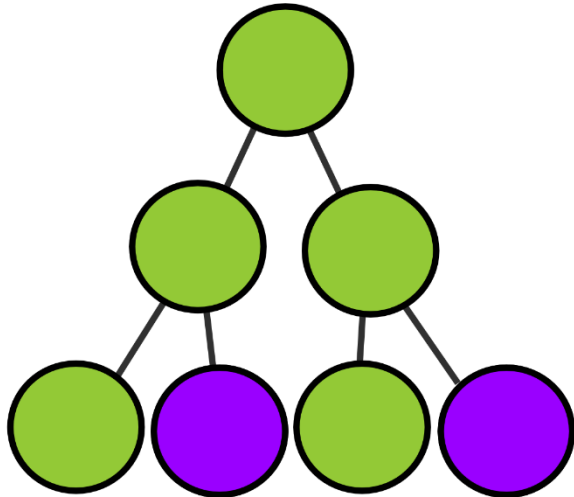


Requête

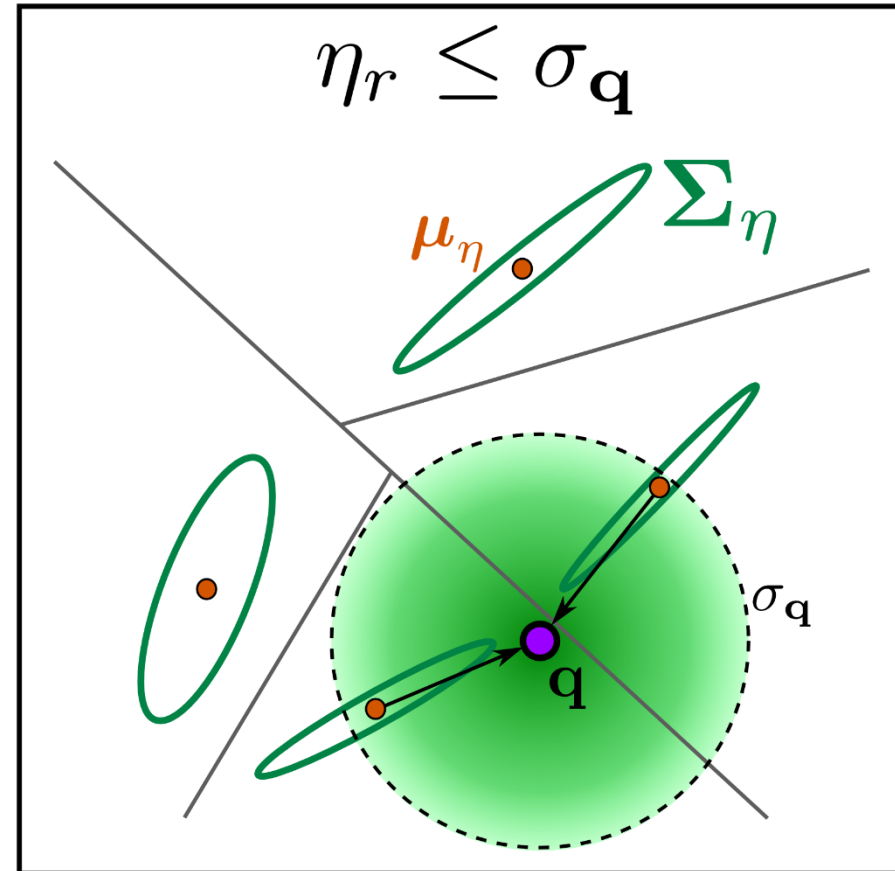
Query

$$\hat{\mu} = \sum_i w_i \mu_{\eta_i}$$

$$\hat{\Sigma} = \sum_i w_i \Sigma_{\eta_i}$$



Bsp-Tree

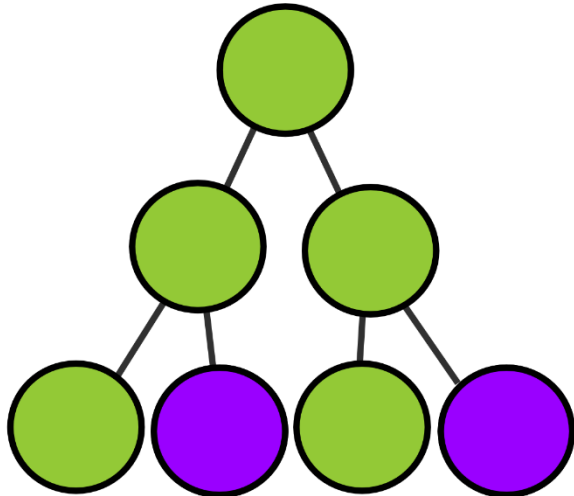


Requête

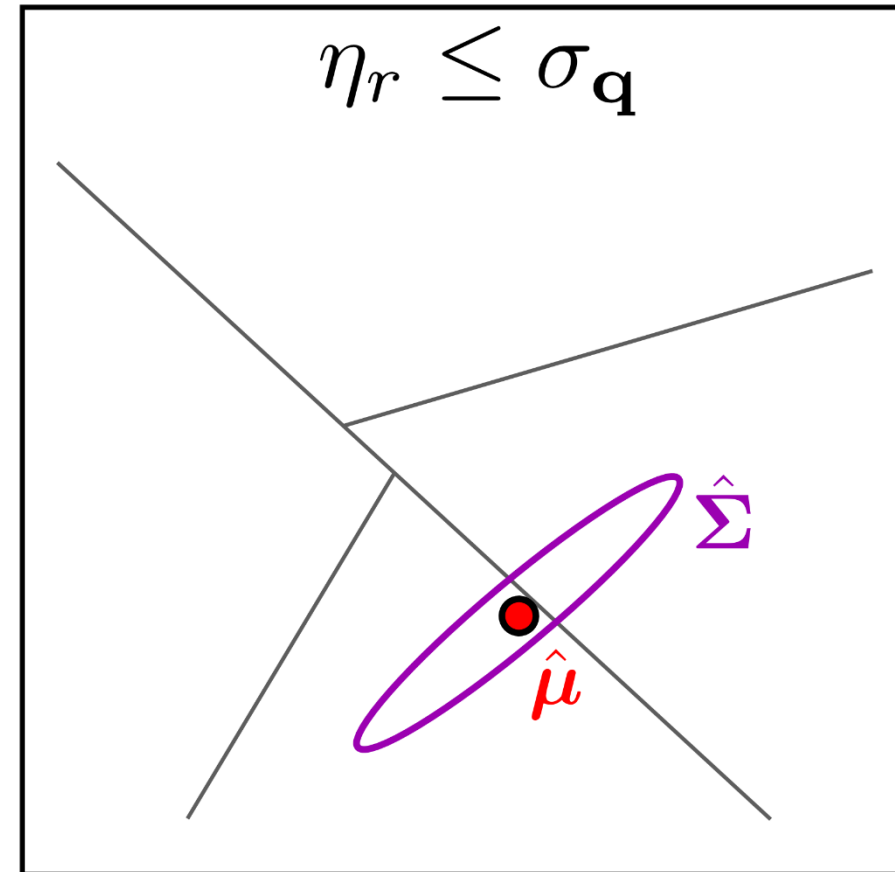
Query

$$\hat{\boldsymbol{\mu}} = \frac{1}{\hat{w}} \sum_i w_i \boldsymbol{\mu}_{\eta_i}$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{\hat{w}_2} \sum_i w_i \boldsymbol{\Sigma}_{\eta_i} - \hat{w} \hat{\boldsymbol{\mu}}^t \hat{\boldsymbol{\mu}}$$

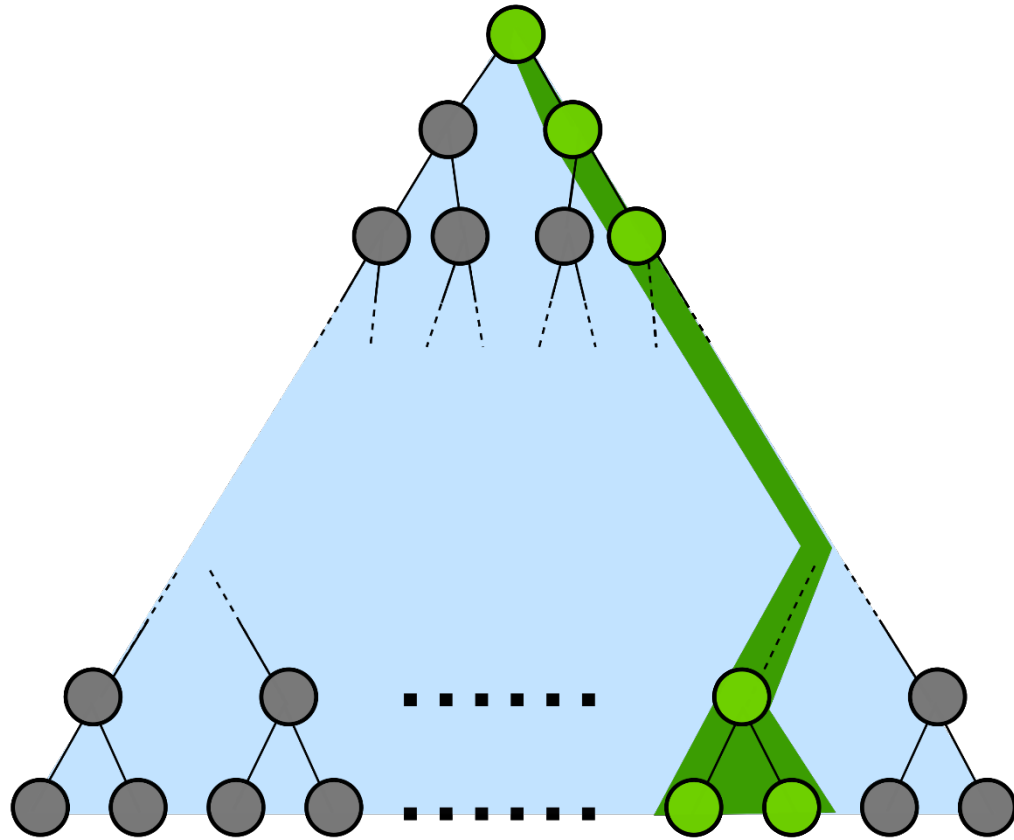


Bsp-Tree

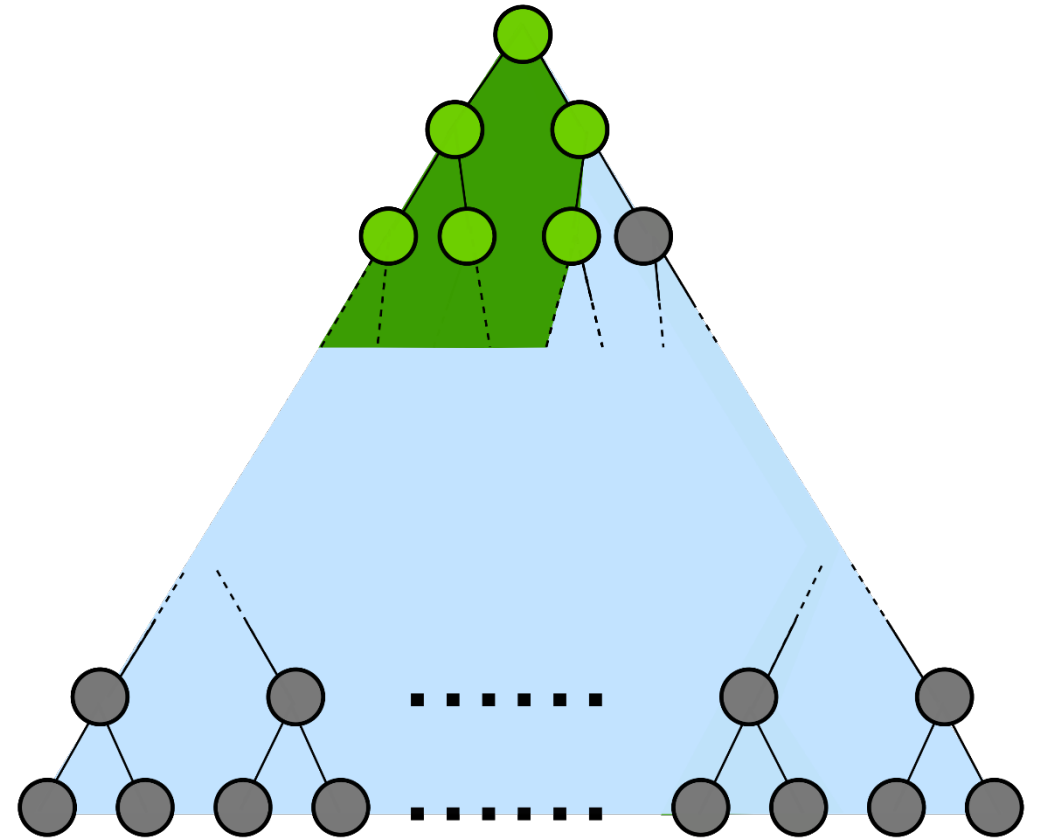


Requête

Computational Performance

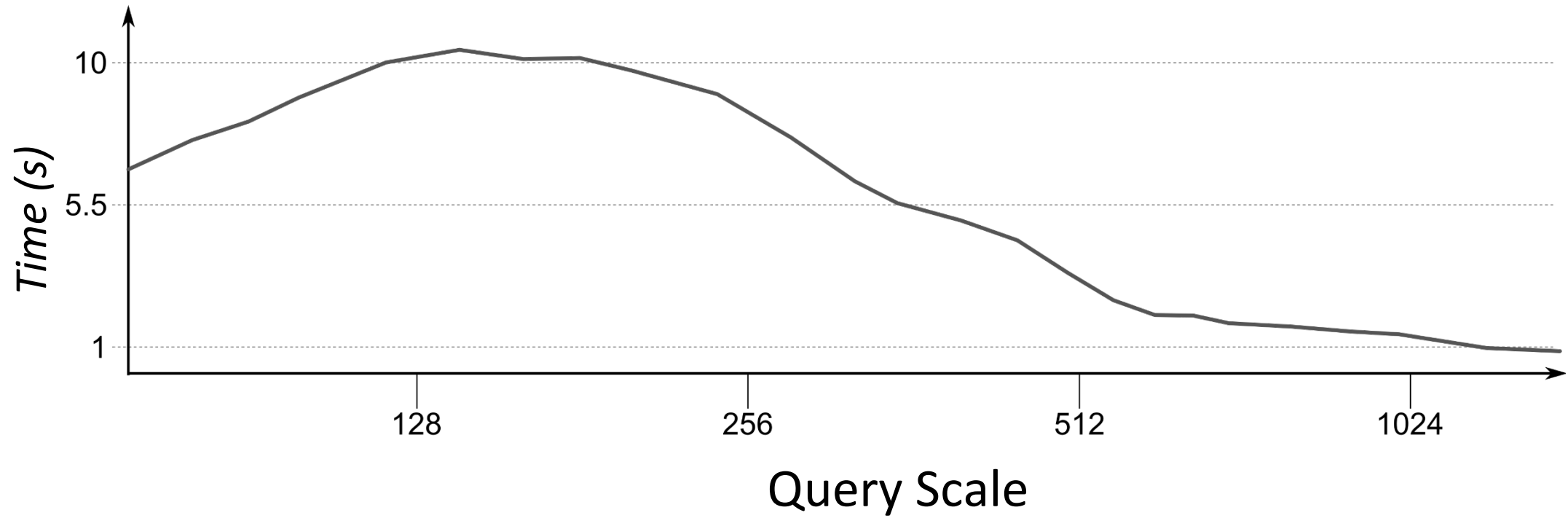


Requête avec échelle fine



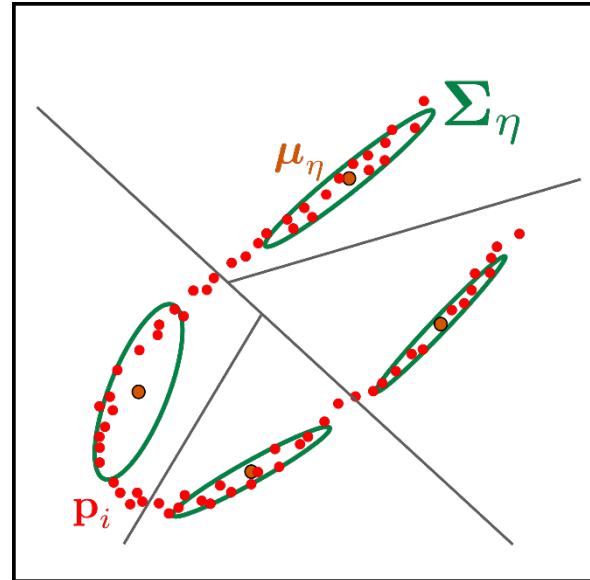
Requête avec échelle large

Computational Performance



Generalisation

Context domain



Value domain

$$\mu_i \propto \sum_i \phi\left(\frac{\mathbf{p}_i - \mathbf{q}}{\sigma_q}\right) \mathbf{f}_i$$

$$\Sigma_i \propto \sum_i \phi\left(\frac{\mathbf{p}_i - \mathbf{q}}{\sigma_q}\right) \mathbf{f}_i \mathbf{f}_i^T$$

Non Local Means [Buades et al. 05]

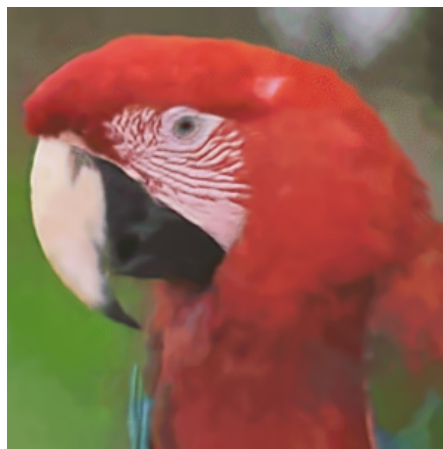
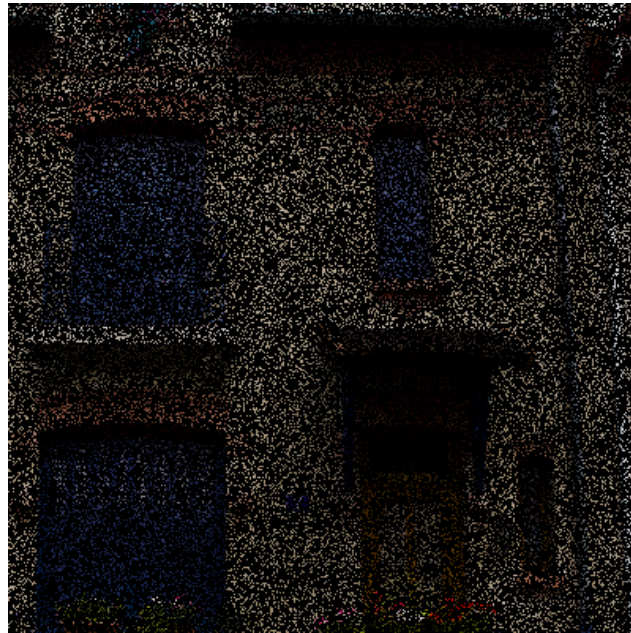


Image Restoration

Applications



Filtrage d'images



Approche par dictionnaire



Restauration de surfaces

Applications



Filtrage d'images



Approche par dictionnaire



Restauration de surfaces

Non Local Bayes [Lebrun et al. 12]

$$\mathbb{P}(\mathbf{q}|\hat{\mathbf{q}}) = \mathcal{N}(\mathbf{0}, \sigma_d)$$

$$\mathbb{P}(\hat{\mathbf{q}}) = \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$$

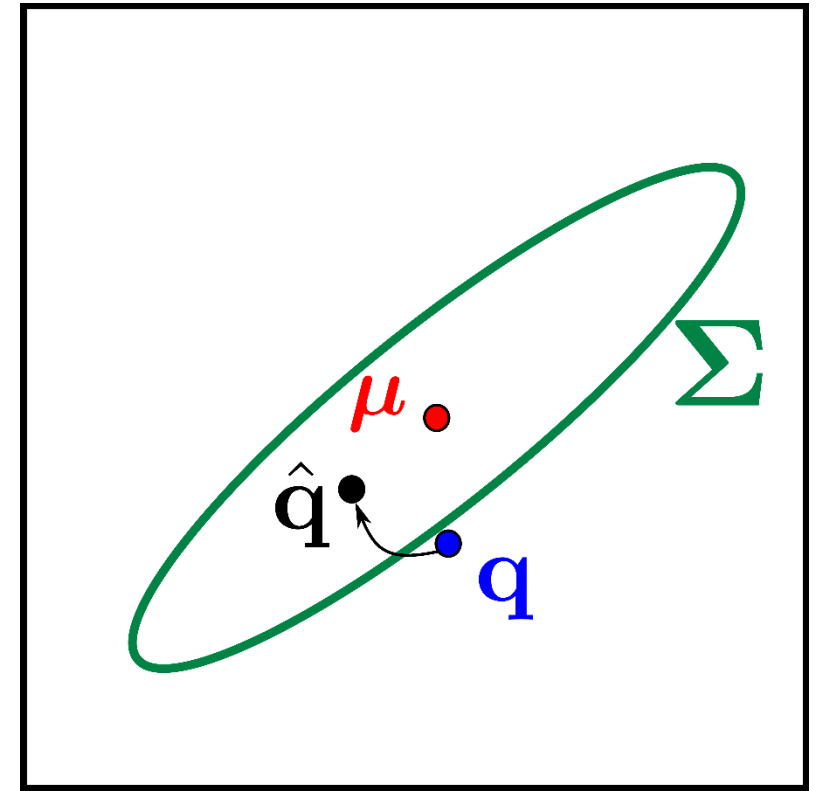
$$\mathbb{P}(\mathbf{q}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \sigma_d^2 Id$$

$$\underset{\hat{\mathbf{q}}}{\operatorname{argmax}} \mathbb{P}(\hat{\mathbf{q}}|\mathbf{q}) = \underset{\hat{\mathbf{q}}}{\operatorname{argmax}} \mathbb{P}(\mathbf{q}|\hat{\mathbf{q}})\mathbb{P}(\hat{\mathbf{q}})$$

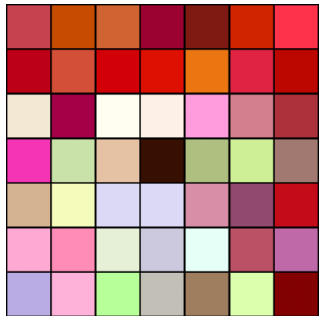
$$\hat{\mathbf{q}} = \boldsymbol{\mu} + [\boldsymbol{\Sigma} - \sigma_d^2 Id] \boldsymbol{\Sigma}^{-1} (\mathbf{q} - \boldsymbol{\mu})$$

Non Local Bayes [Lebrun et al. 12]

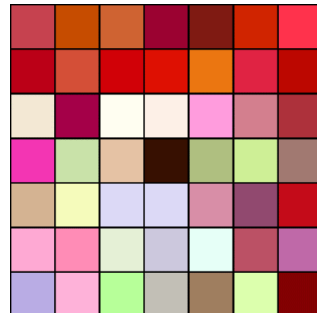
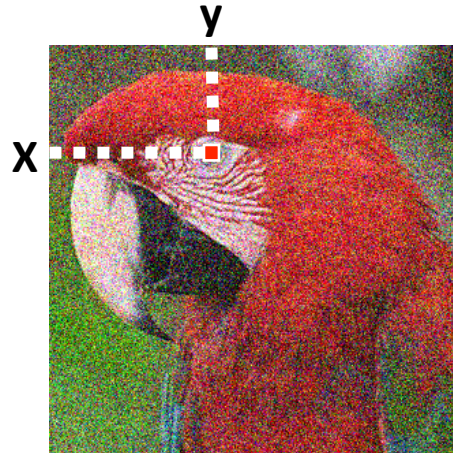


$$\hat{q} = \mu + [\Sigma - \sigma_d^2 Id] \Sigma^{-1} (q - \mu)$$

Different grouping criteria

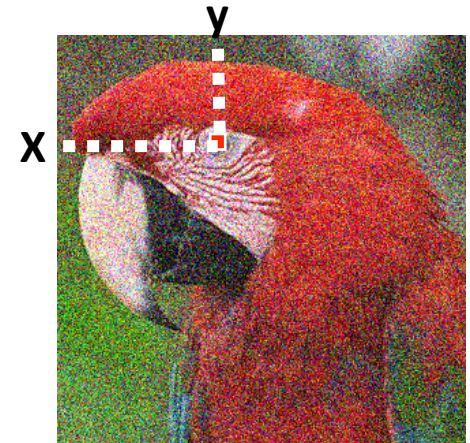


Global (3x7x7 – D)



+ (X, Y)

Local (3x7x7+2 – D)

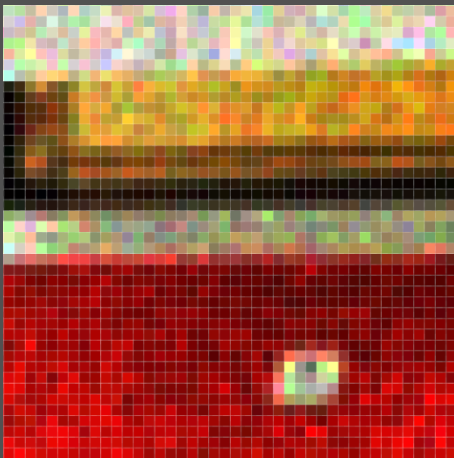


6-D principales + (X, Y)

Compact (6 – D)

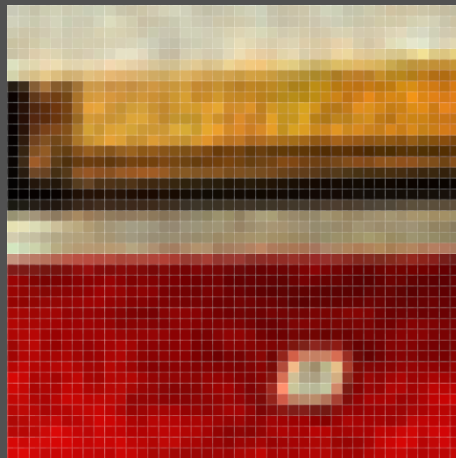
Results: Different grouping criteria

Noisy



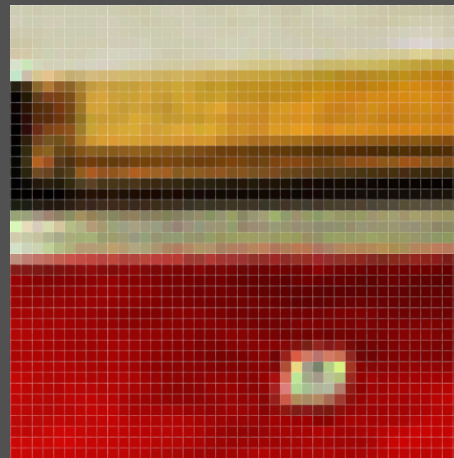
PSNR : 22.7 dB

Global



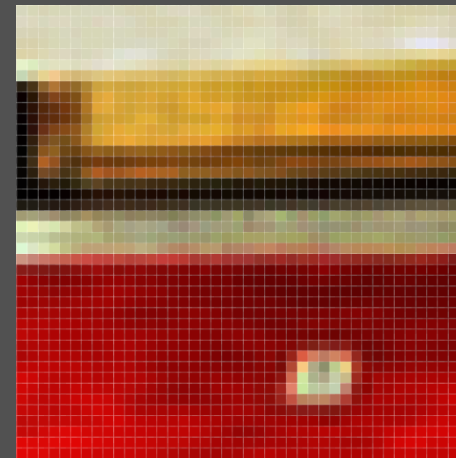
PSNR : 27.6 dB

Local



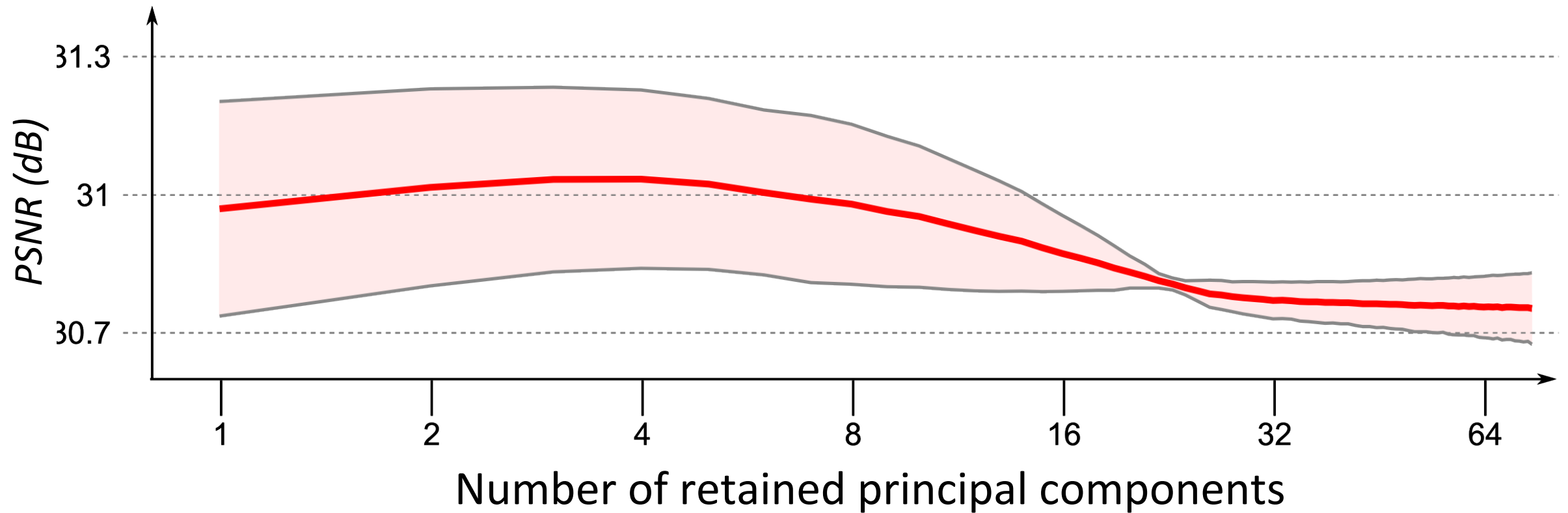
PSNR : 28.0 dB

Compact



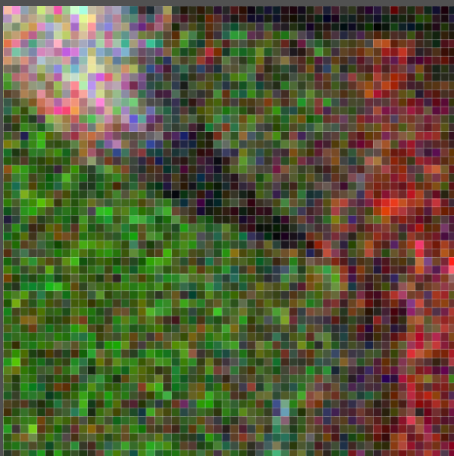
PSNR : 28.4 dB

Compressed grouping criterion



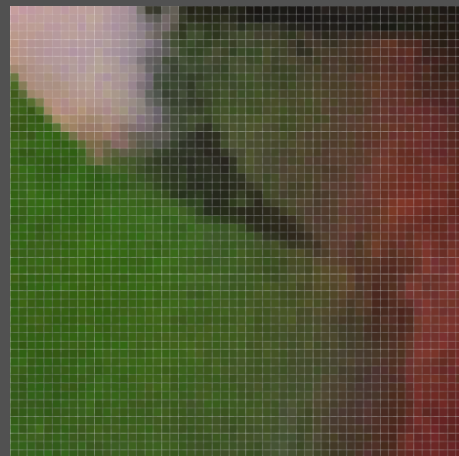
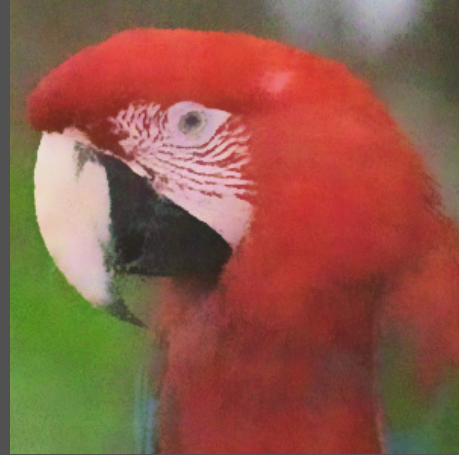
Results: Non Local Bayes

Noisy



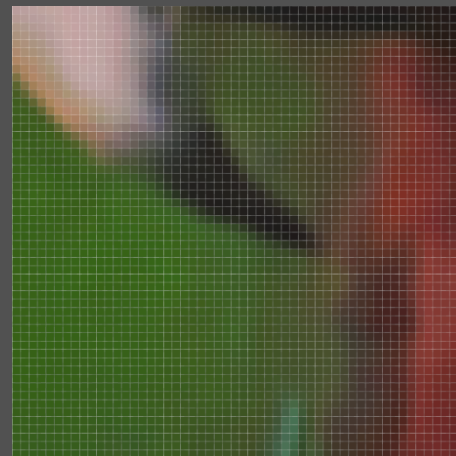
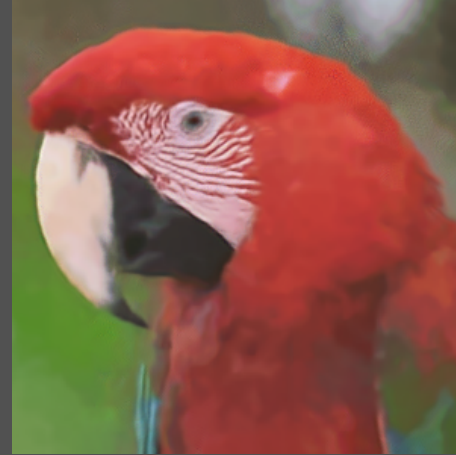
PSNR : 14.7dB

NL-Means



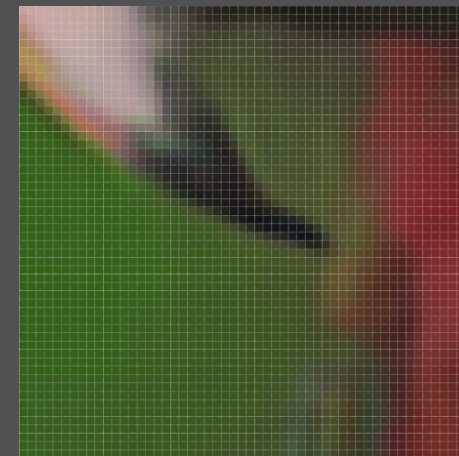
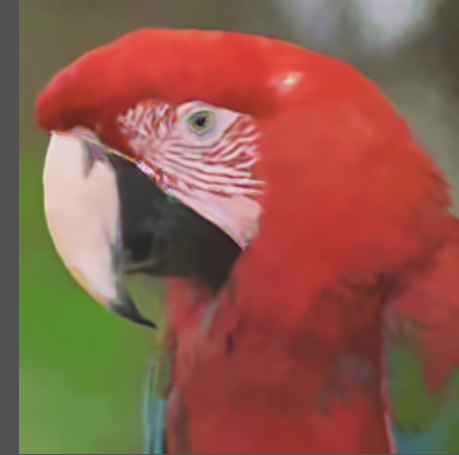
PSNR : 27.9 dB
[Buades et al. 05]

NLB



PSNR : 29.0 dB
[Lebrun et al. 12]

CovTree NLB

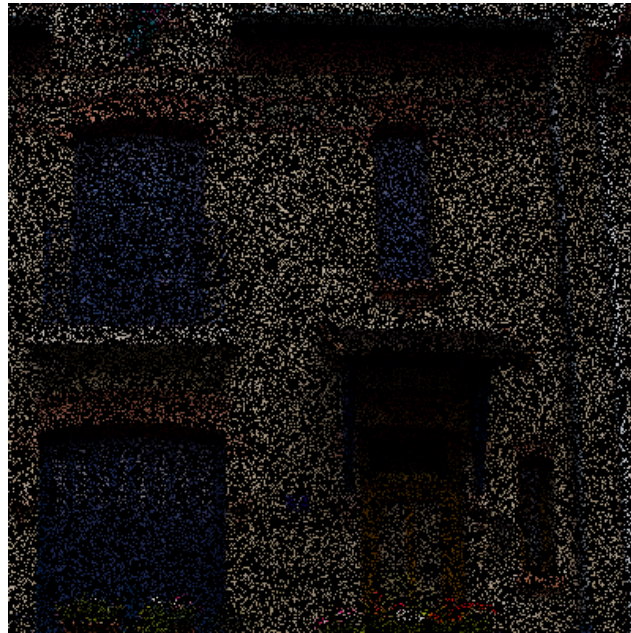


PSNR : 29.2 dB

Applications



Filtrage d'images



Approche par dictionnaire



Restauration de surfaces

Learning-based denoising

- Data-base
 - 120 images
 - 100 000 000 patches
- Learning complexity
 - Building: 5 minutes
 - Learning: 5 hours
- Query:
 - 1 million pixels
 - Query: 30 seconds



Learning-based denoising

$$\mathbb{P}(\mathbf{q}|\hat{\mathbf{q}}) = \mathcal{N}(\mathbf{0}, \sigma_d)$$

$$\mathbb{P}(\hat{\mathbf{q}}) = \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$$

$$\mathbb{P}(\mathbf{q}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} - \sigma_d^2 Id$$

$$\underset{\hat{\mathbf{q}}}{\operatorname{argmax}} \mathbb{P}(\hat{\mathbf{q}}|\mathbf{q}) = \underset{\hat{\mathbf{q}}}{\operatorname{argmax}} \mathbb{P}(\mathbf{q}|\hat{\mathbf{q}})\mathbb{P}(\hat{\mathbf{q}})$$

$$\hat{\mathbf{q}} = \boldsymbol{\mu} + \boldsymbol{\Sigma}[\boldsymbol{\Sigma} + \sigma_d^2 Id]^{-1}(\mathbf{q} - \boldsymbol{\mu})$$

Learning-based denoising

Bruité



PSNR : 22.4 dB

NLB



PSNR : 30.2 dB
[Lebrun et al. 12]

CovTree NLB



PSNR : 30.0 dB

CovTree + Dictionnary



PSNR : 31.1 dB

Reconstruction from Random Samples

$$\tilde{\mathbf{q}} = S\mathbf{q}$$

$$\hat{\mathbf{q}} = (\Sigma_{\mathbf{q}} S^H S + \frac{\sigma^2}{2} Id)^{-1} (\Sigma_{\mathbf{q}} S^H \tilde{\mathbf{q}} + \frac{\sigma^2}{2} \boldsymbol{\mu}_{\mathbf{q}})$$

$$\hat{\mathbf{q}}_t = (\Sigma_{\hat{\mathbf{q}}_{t-1}} S^H S + \frac{\sigma^2}{2} Id)^{-1} (\Sigma_{\hat{\mathbf{q}}_{t-1}} S^H \tilde{\mathbf{q}} + \frac{\sigma^2}{2} \boldsymbol{\mu}_{\hat{\mathbf{q}}_{t-1}})$$

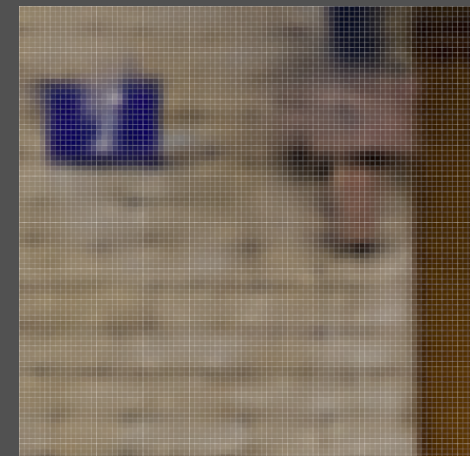
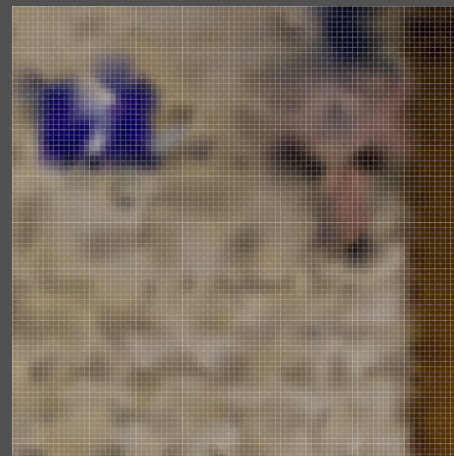
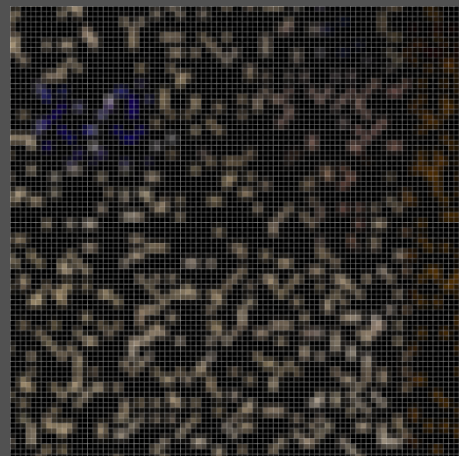
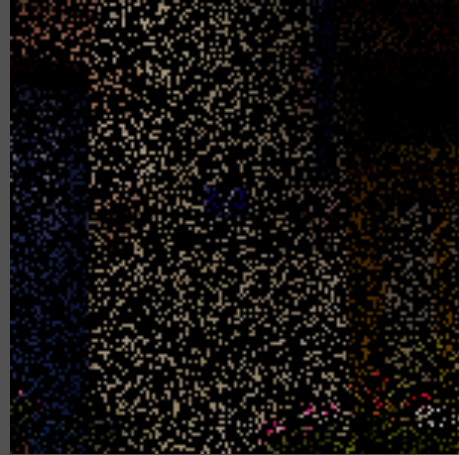
Reconstruction d'image échantillonnée aléatoirement

Original

20% pixel originaux

Cubique

CovTree - Dictionnaire



PSNR : 24.9 dB

PSNR : 27.2 dB

Applications



Filtrage d'images

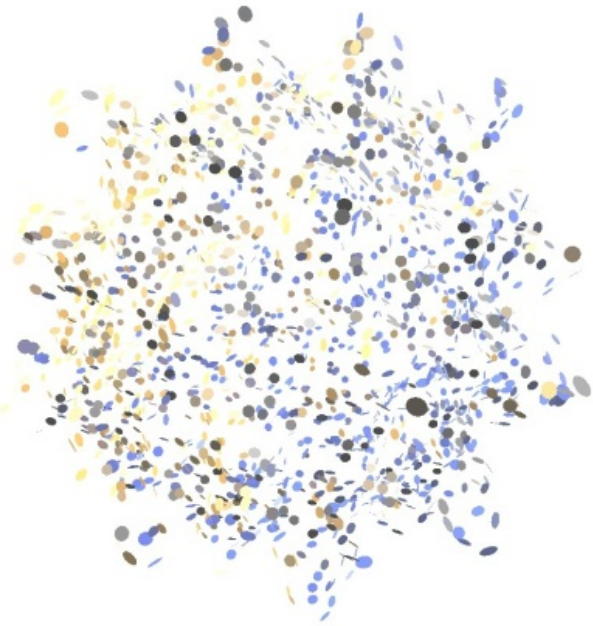


Approche par dictionnaire

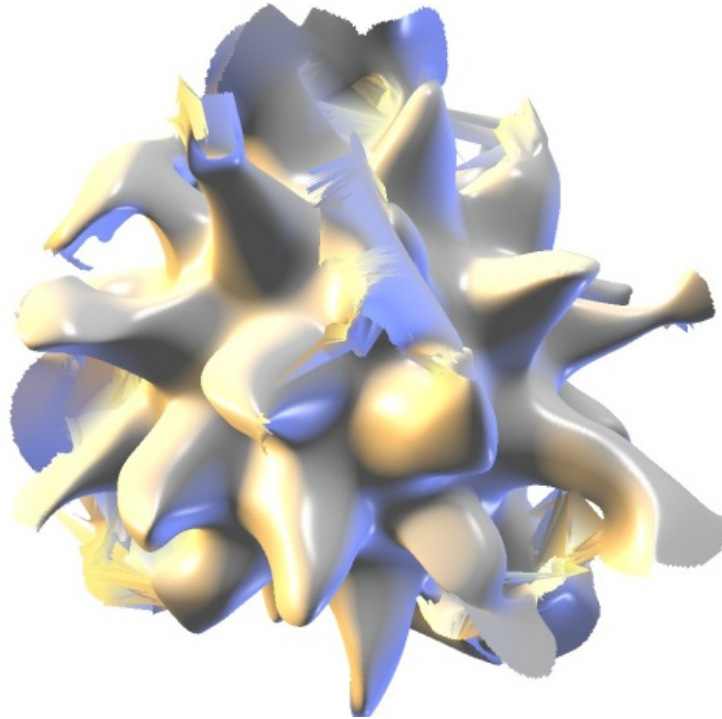


Restauration de surface

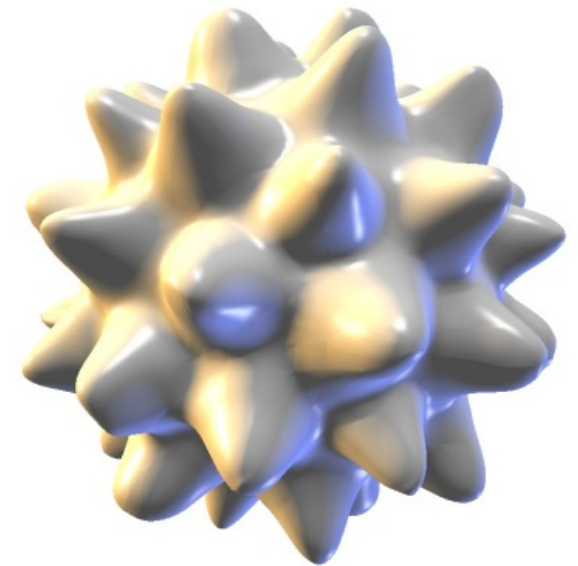
Surface reconstruction



3D point cloud



Point Set Surfaces
RIMLS [Öztireli et al. 2009]

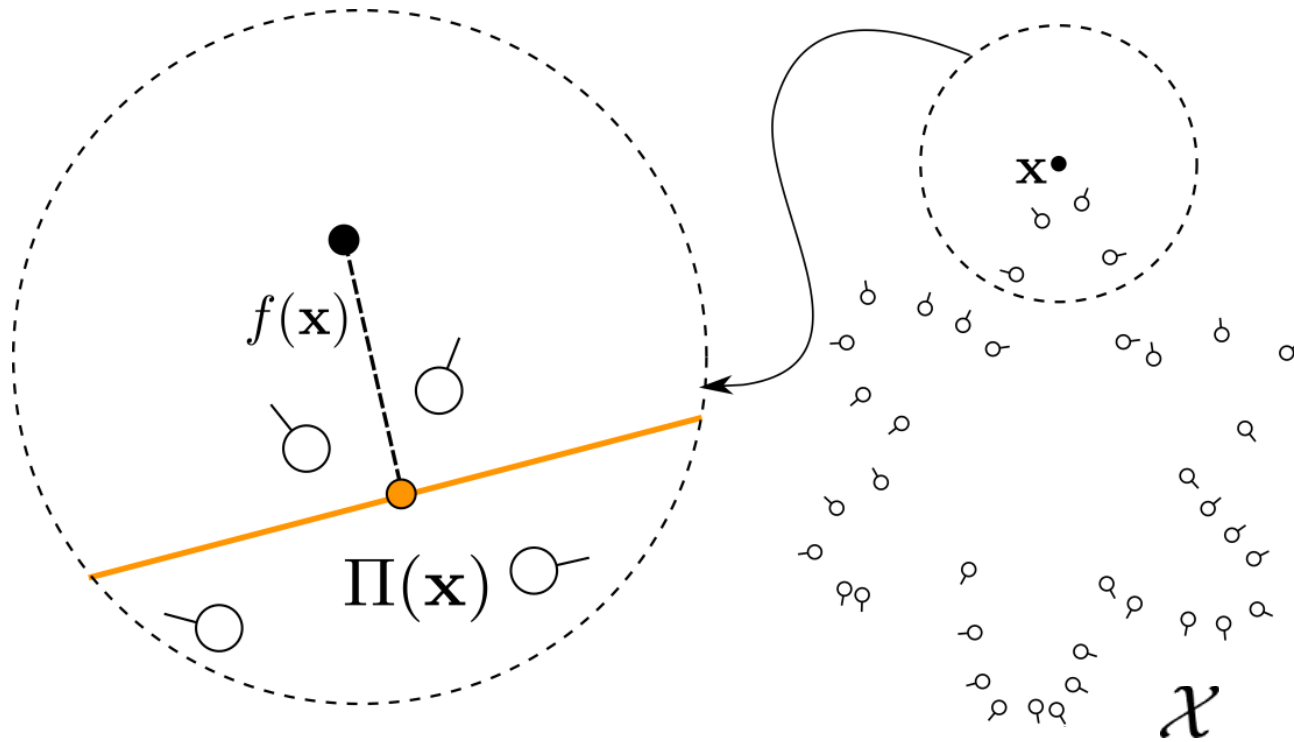


Non-local Point Set Surface

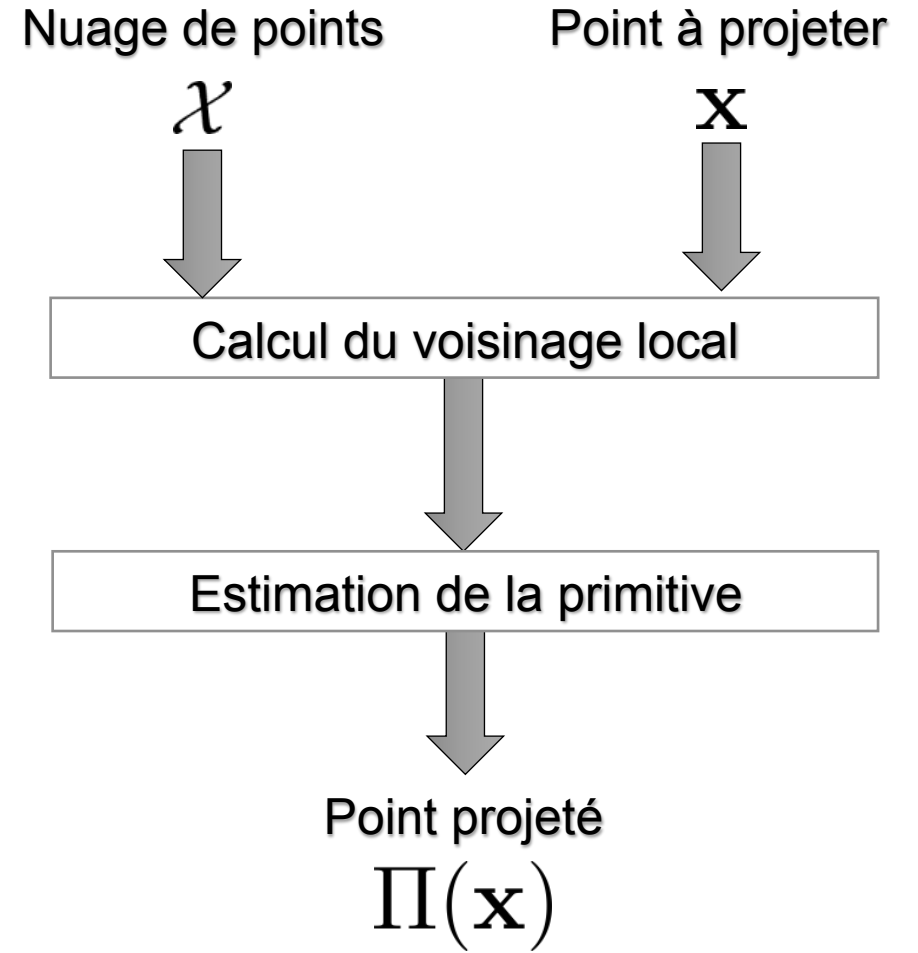
Surface reconstruction



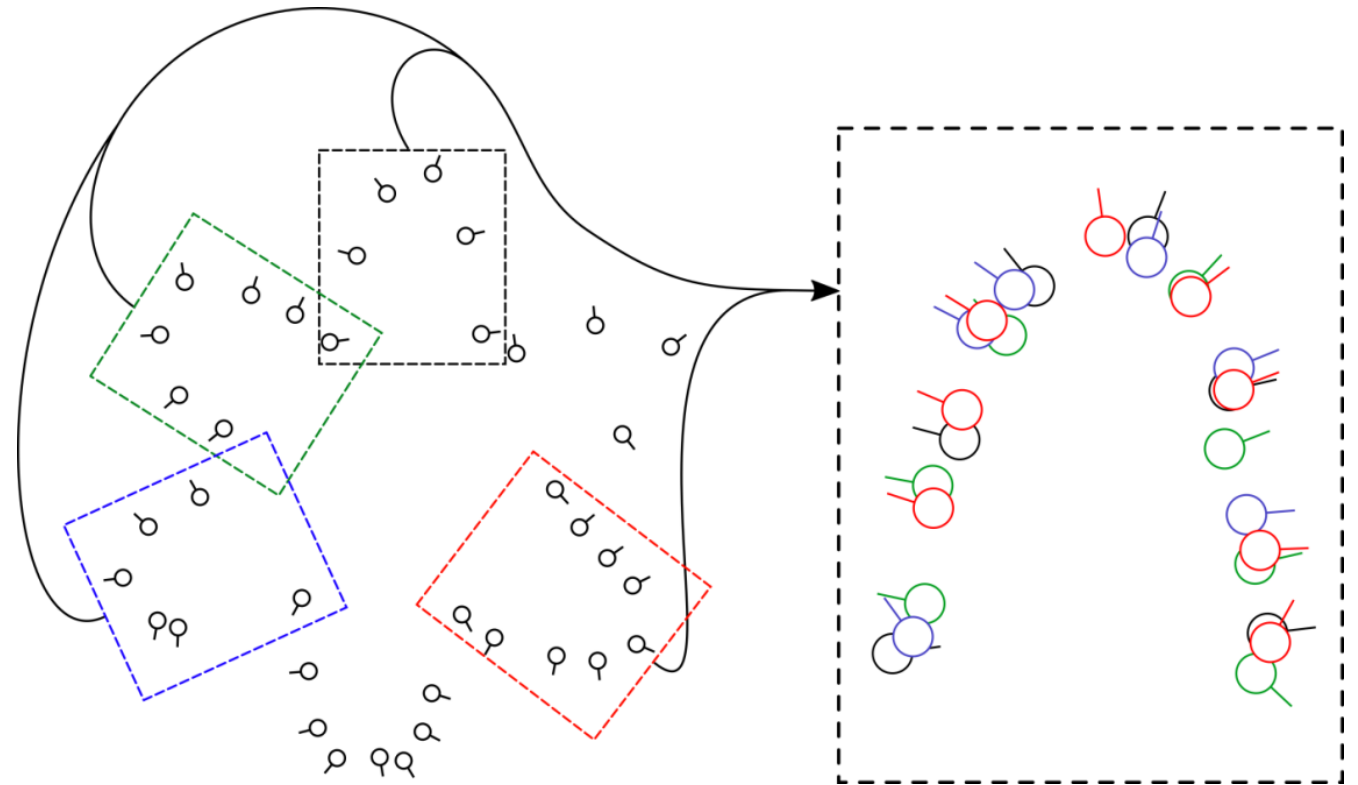
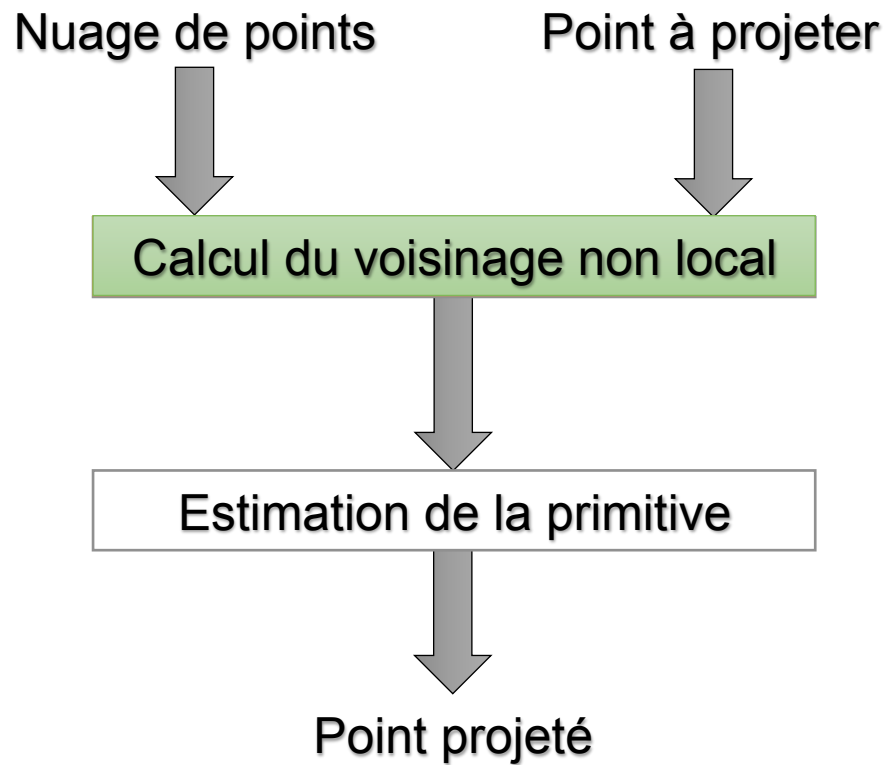
Point Set Surfaces – PSS



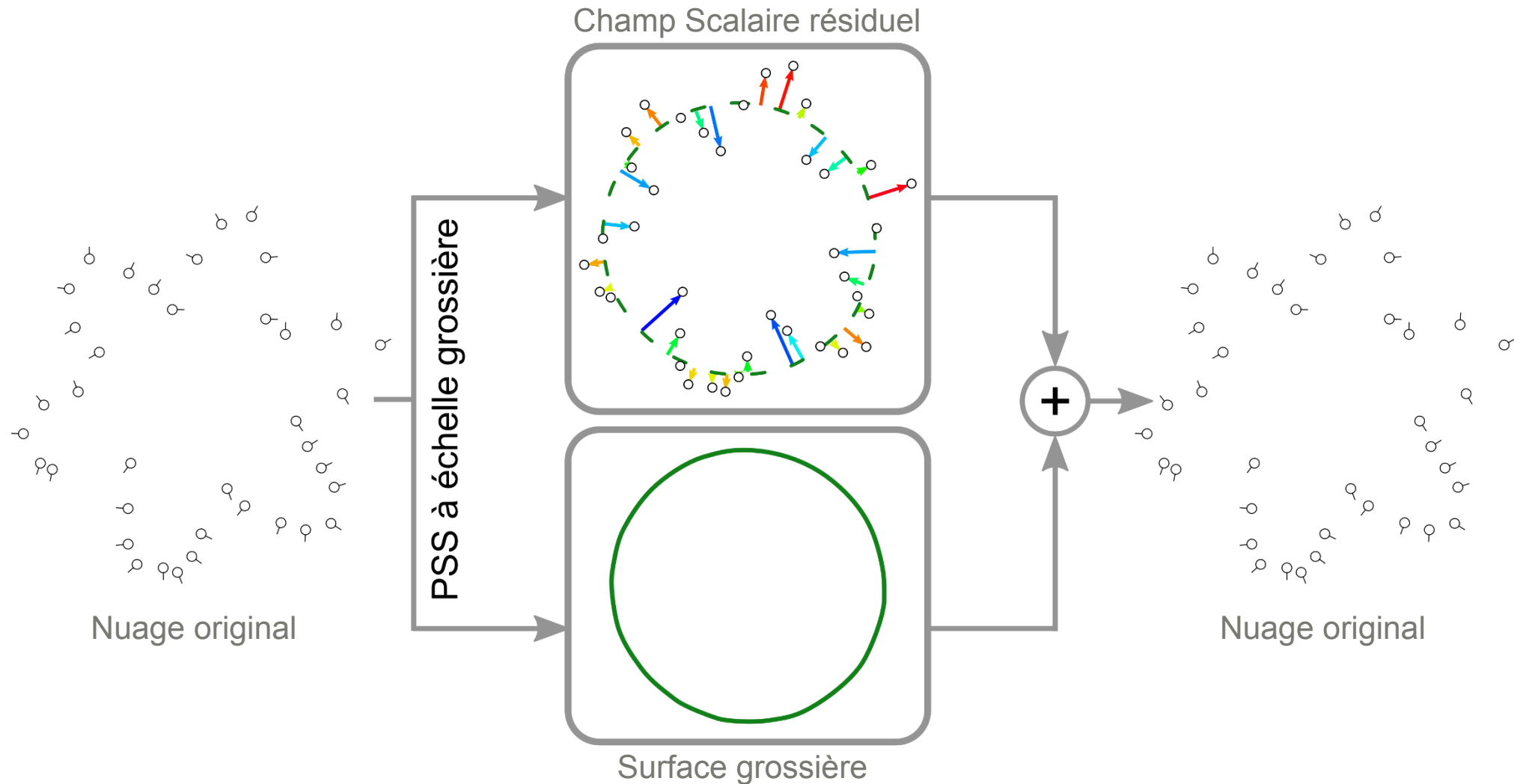
$$\Pi(\mathbf{x}) = \mathbf{x} - f(\mathbf{x}) \mathbf{n}(\mathbf{x})$$



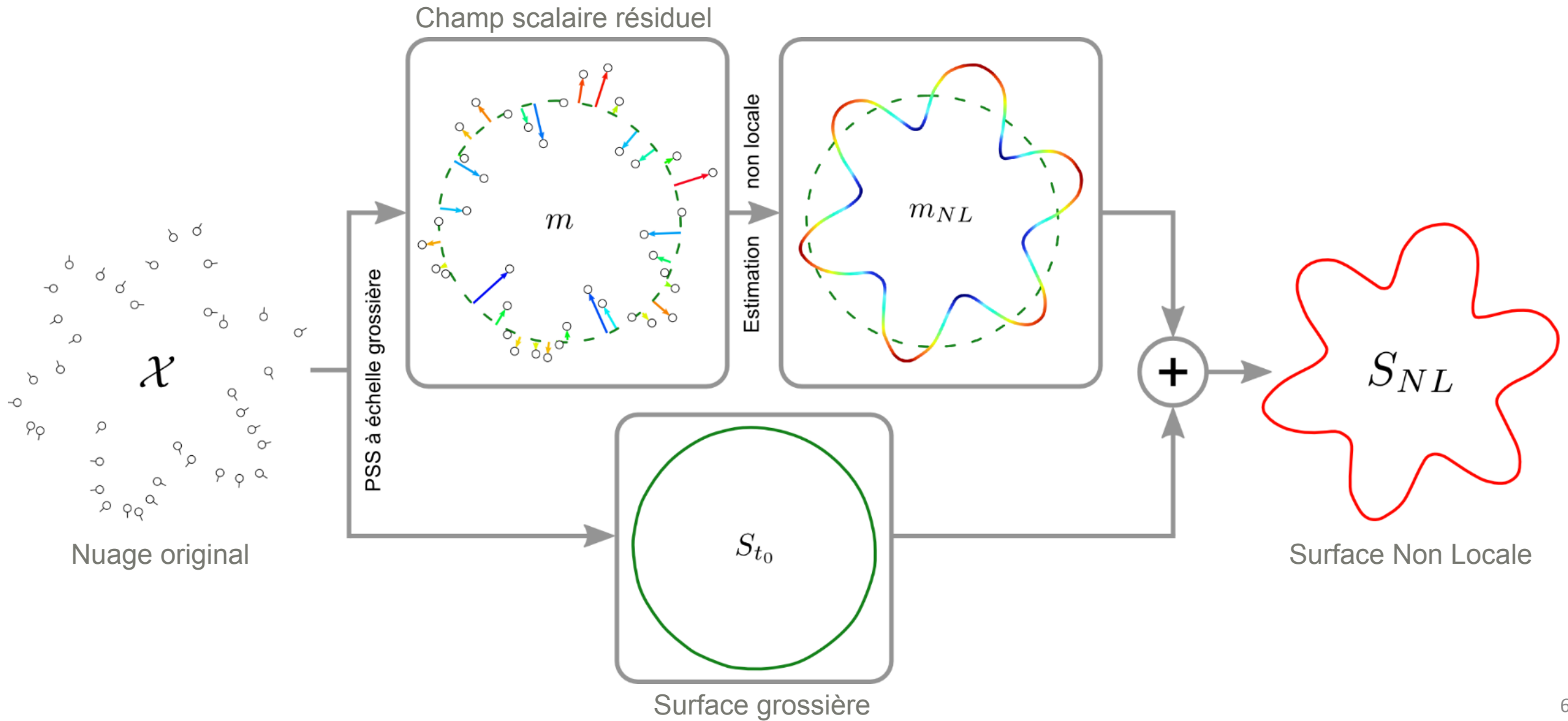
Non Local vs Local denoising



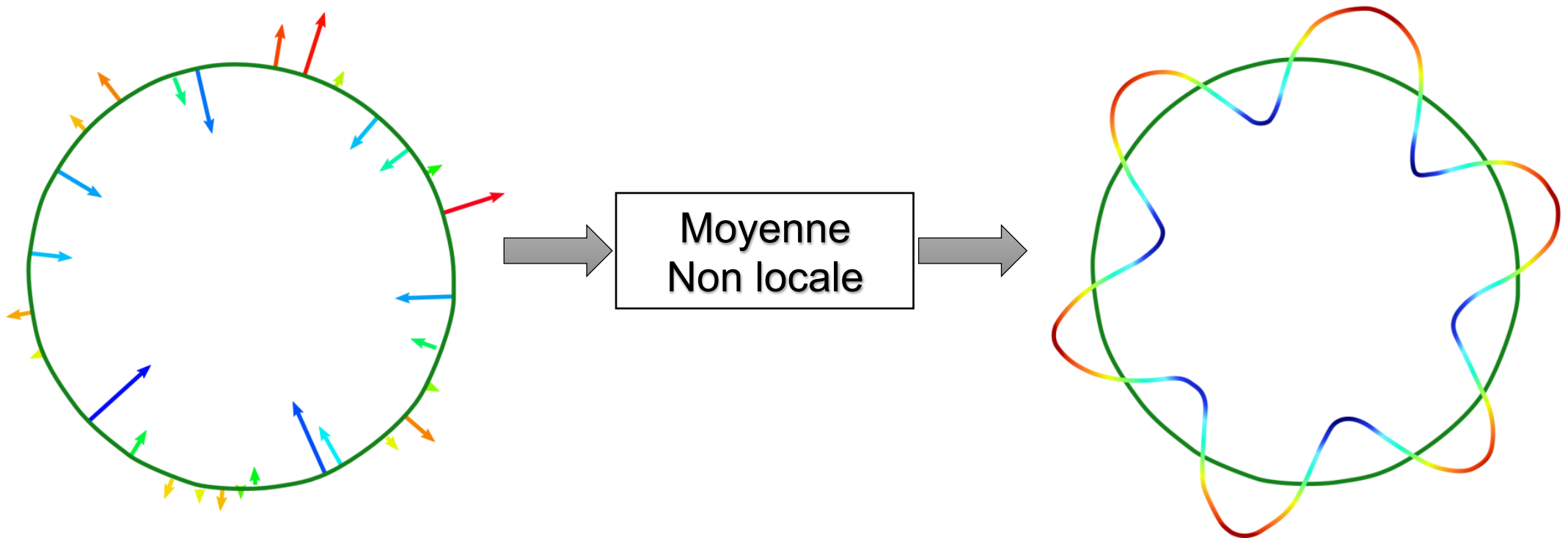
PSS decomposition



PSS decomposition + interpolation

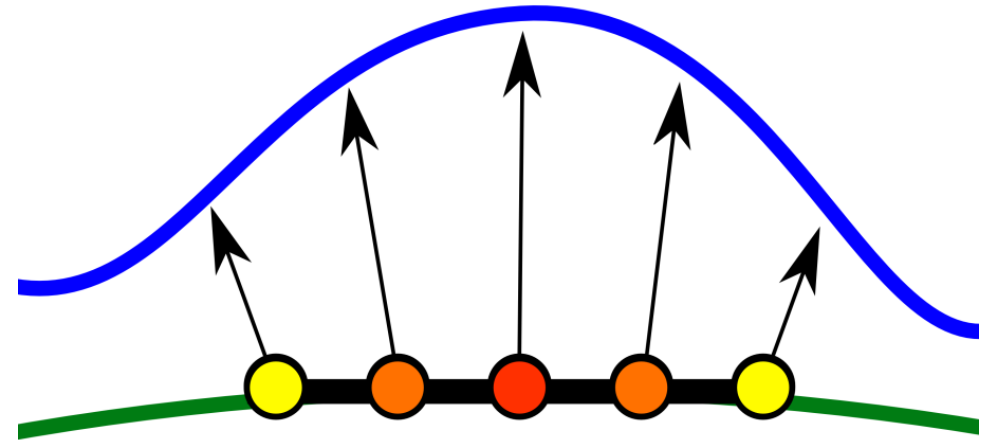
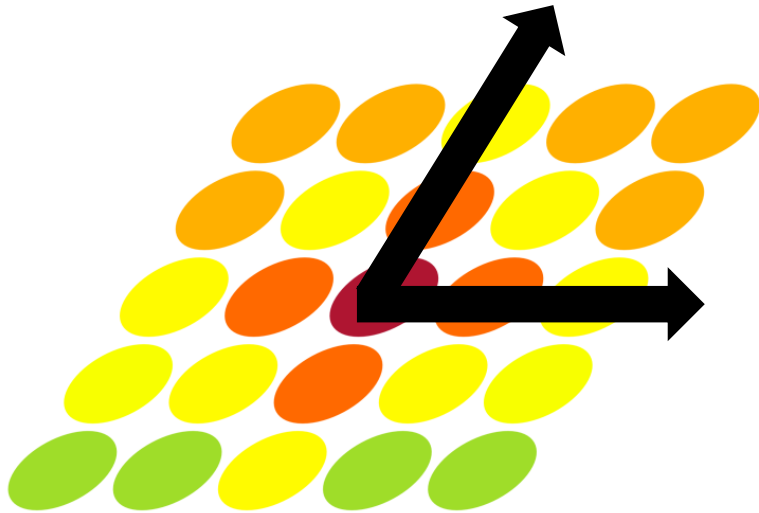


Non-local interpolation



$$m_{NL}(\mathbf{x}) = \sum_{\mathbf{p}_i \in \mathcal{X}} w_{NL}(\mathbf{x}, \mathbf{p}_i) m(\mathbf{p}_i)$$

Local surface descriptor: 2D patch of scalar displacements

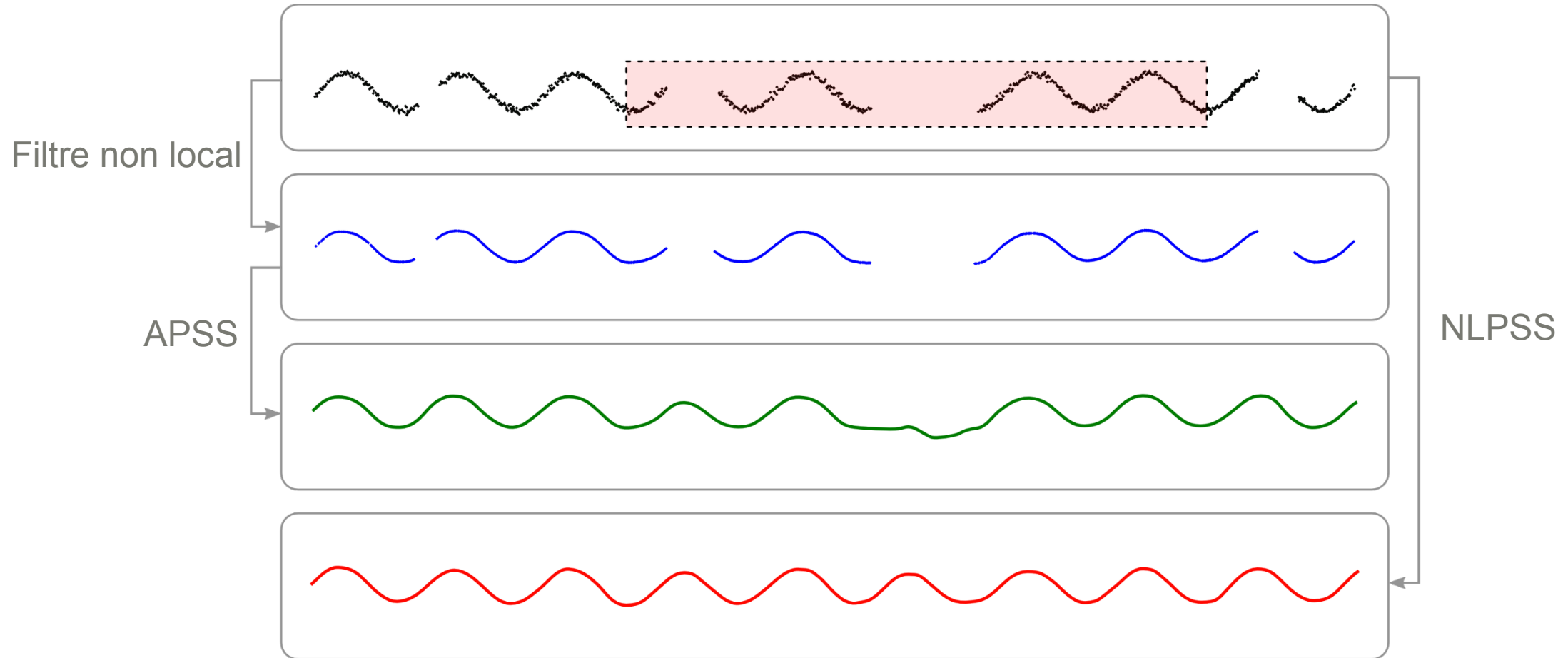


L2 similarity measure

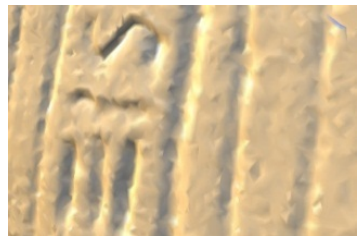
$$d^2 = \left\| \begin{array}{ccccc} \text{orange} & \text{orange} & \text{yellow} & \text{orange} & \text{orange} \\ \text{orange} & \text{yellow} & \text{orange} & \text{yellow} & \text{orange} \\ \text{yellow} & \text{orange} & \text{red} & \text{orange} & \text{yellow} \\ \text{yellow} & \text{yellow} & \text{orange} & \text{yellow} & \text{yellow} \\ \text{green} & \text{green} & \text{yellow} & \text{green} & \text{green} \end{array} - \begin{array}{ccccc} \text{orange} & \text{orange} & \text{yellow} & \text{orange} & \text{orange} \\ \text{orange} & \text{green} & \text{blue} & \text{green} & \text{green} \\ \text{yellow} & \text{blue} & \text{blue} & \text{blue} & \text{green} \\ \text{green} & \text{green} & \text{blue} & \text{green} & \text{green} \\ \text{green} & \text{green} & \text{green} & \text{green} & \text{green} \end{array} \right\|^2$$

$$w_{NL}(\mathbf{y}, \mathbf{z}) = \frac{1}{Z_{NL}} e\left(-\frac{d^2}{h^2}\right)$$

Non local vs Local



Results: Ramsès Statue



Original

SPSS

[Alexa et al. 04]

HPSS

[Alexa et al. 09]

APSS

[Guennebaud et al. 07]

RIMLS

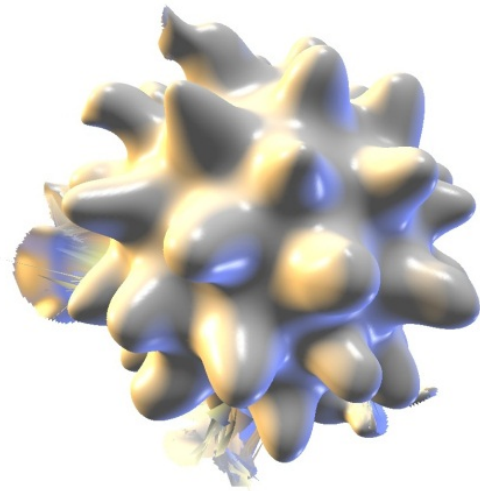
[Öztireli et al. 09]

NLPSS

Results: 3D star

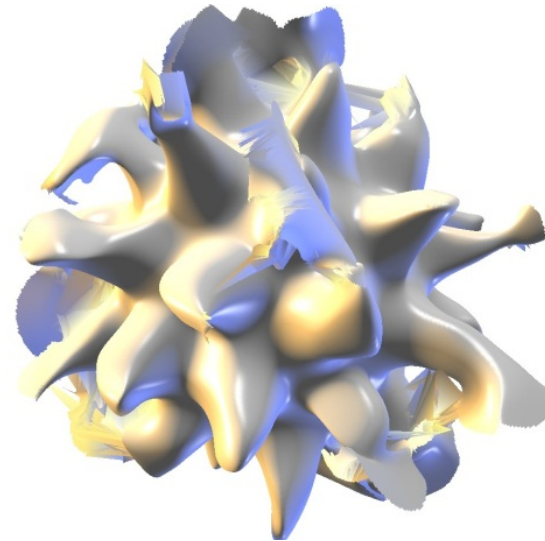


Original



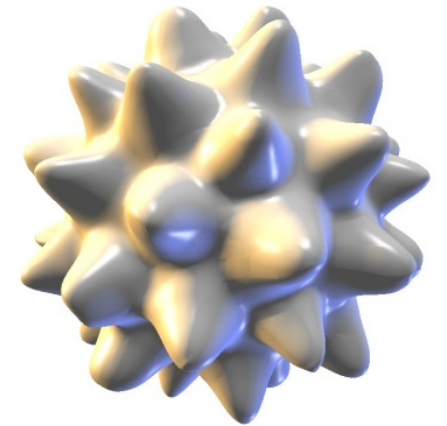
APSS

[Guennebaud et al. 07]



RIMLS

[Öztireli et al. 09]

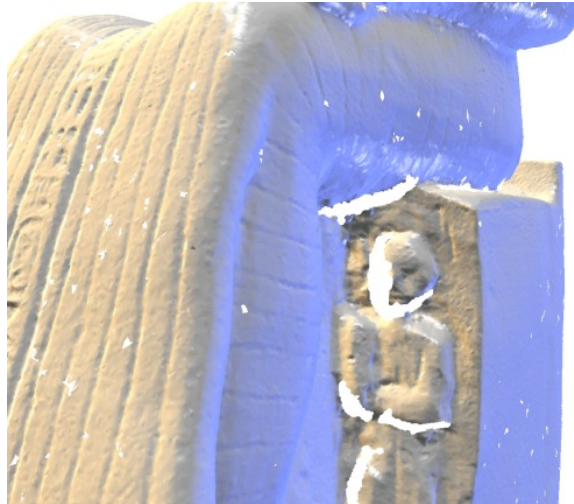


NLPSS

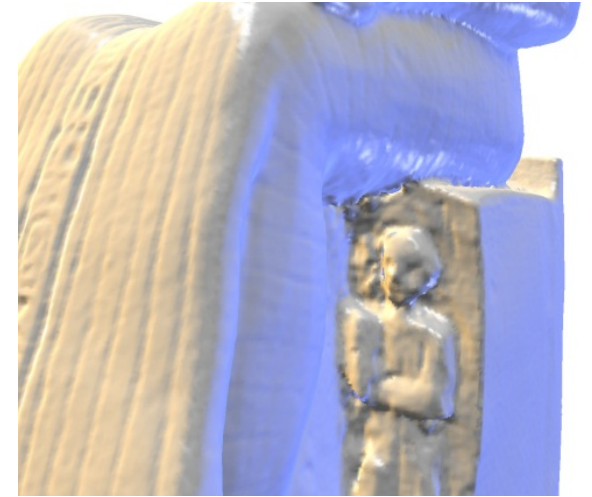
Results: Ramses Statue



Original

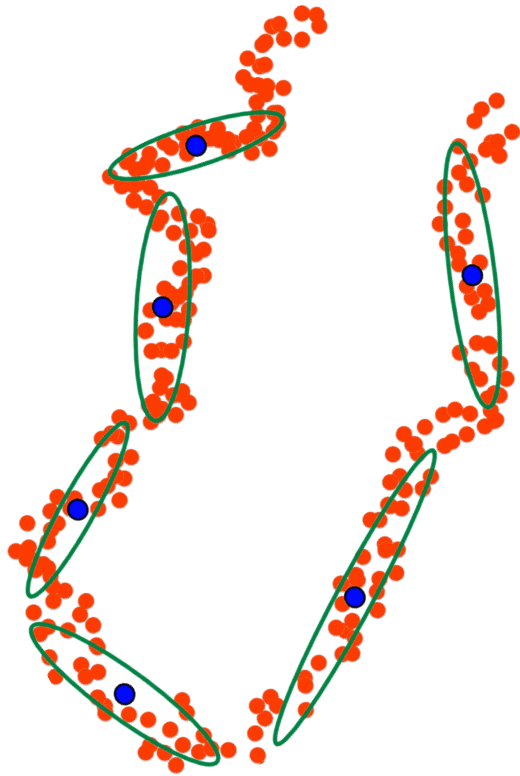


NLPSS



Conclusions

Contributions



Multiscale Tree of
Local Gaussian Models



Denoising,
Random Sampling



Non-local surface
reconstruction

Perspectives

- Multi-resolution (patch size)
- Optimize building step
 - Hierarchical EM
 - On line learning, data stream
- Robust distances, geodesic distances

Thank you for your attention!

Références

Non Local Point Set Surfaces

Thierry Guillemot, Andrés Almansa et Tamy Boubekeur

3DIMPVT – 3D Imaging Modeling Processing Visualization Transmission – 2012

Covariance Trees for 2D and 3D Processing

Thierry Guillemot, Andrés Almansa et Tamy Boubekeur

CVPR – Computer Vision and Pattern Recognition – 2014