LaBRI

Présentation scientifique

EVADoMe

Romain Bourqui

cnrs

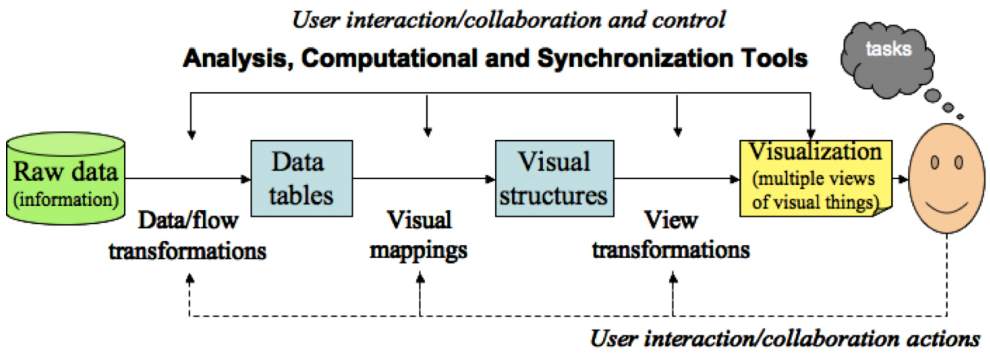
université de BORDEAUX

Bordeaux INP AQUITAINE

EVADoMe : Axes de recherche

Axes de recherche :

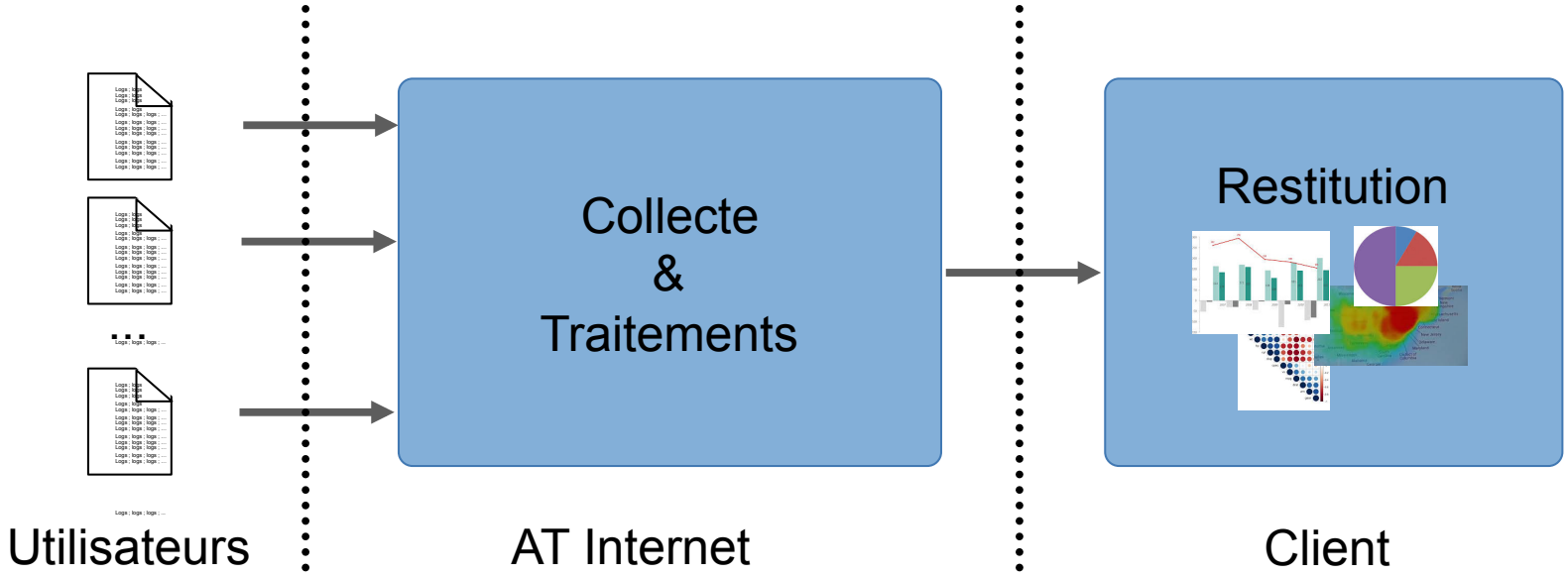
- Fouille de données
- Représentation de données
- Exploration interactive



EVADoMe : Illustration de travaux

Contexte :

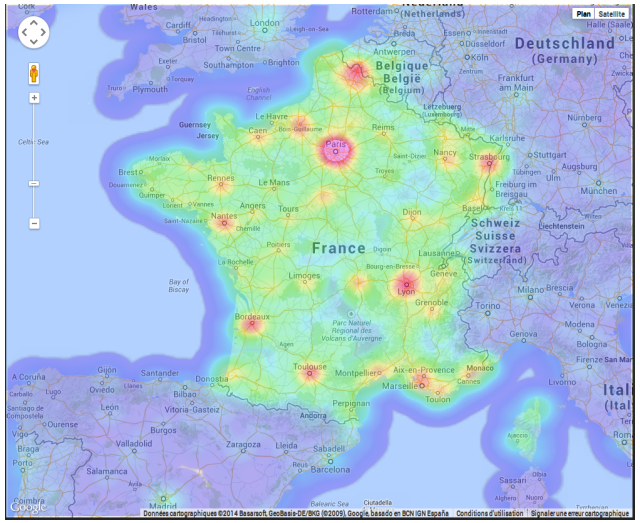
Projet : PIA « Cloud Computing » SpeedData
Entreprise partenaire : AT Internet
Domaine d'activité : *web analytics*



EVADoMe : Illustration de travaux

Application :
Visualisation de Données Géolocalisées

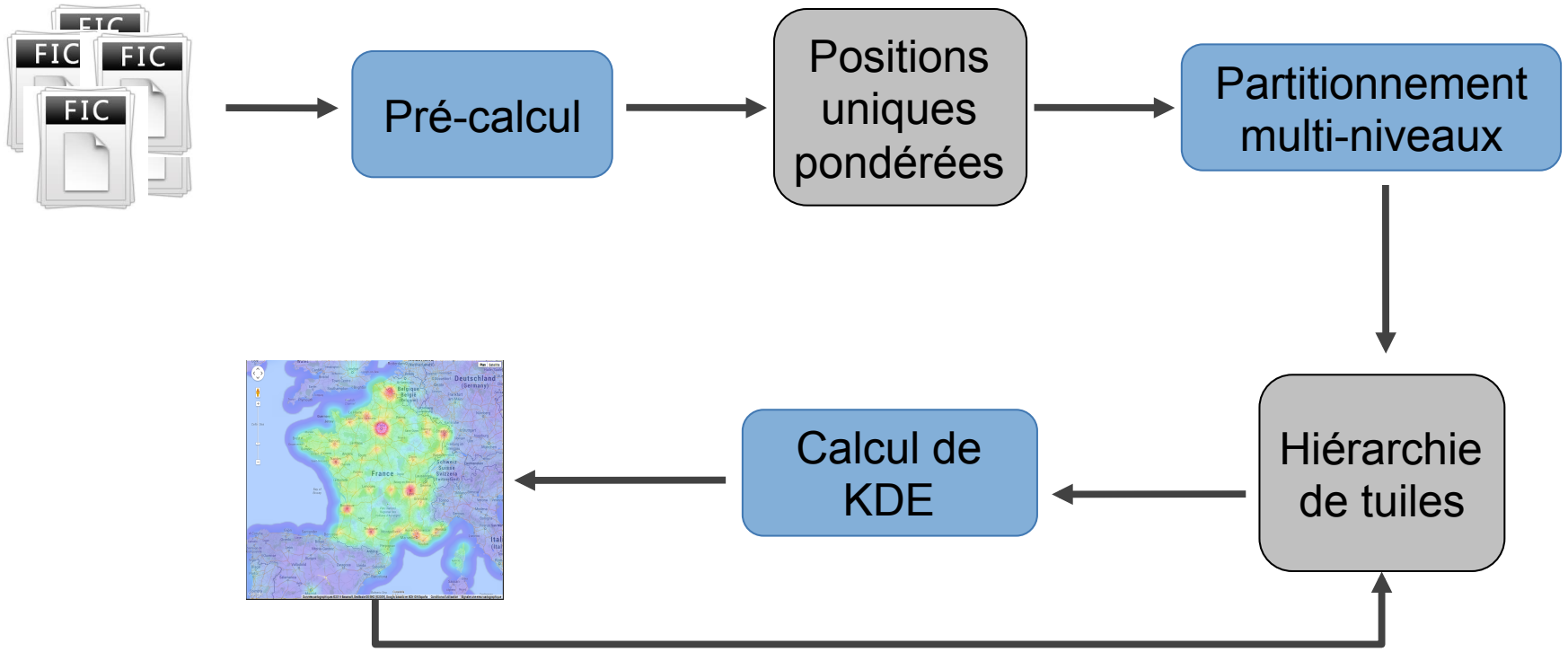
Millions/milliards de logs utilisateurs géolocalisés



Carte de chaleur interactive

EVADoMe : Illustration de travaux

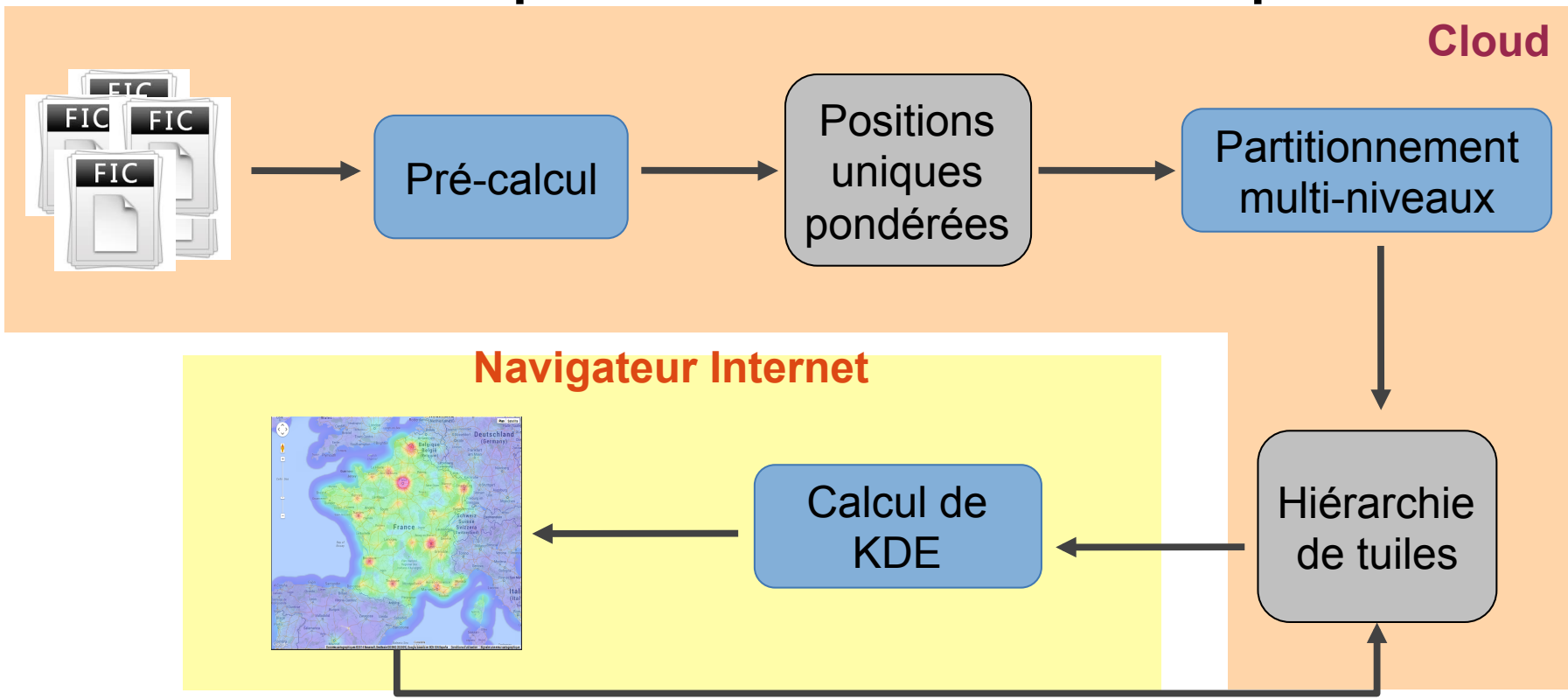
Contribution :
Infrastructure et processus de traitement complet



Interaction utilisateur

EVADoMe : Illustration de travaux

Contribution :
Infrastructure et processus de traitement complet

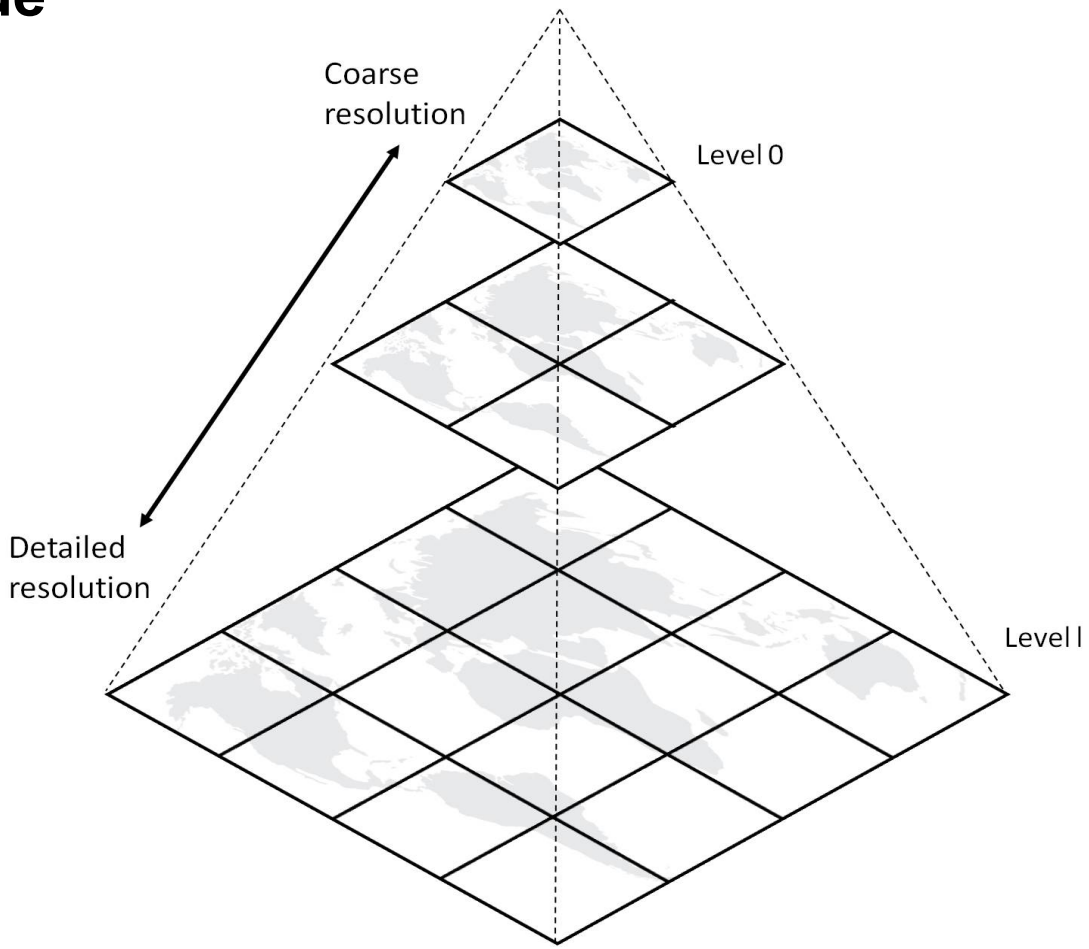


Interaction utilisateur

EVADoMe : Illustration de travaux

Principe de la méthode

Utilisation d'une hiérarchie de tuiles montrant les données avec plus ou moins de précision

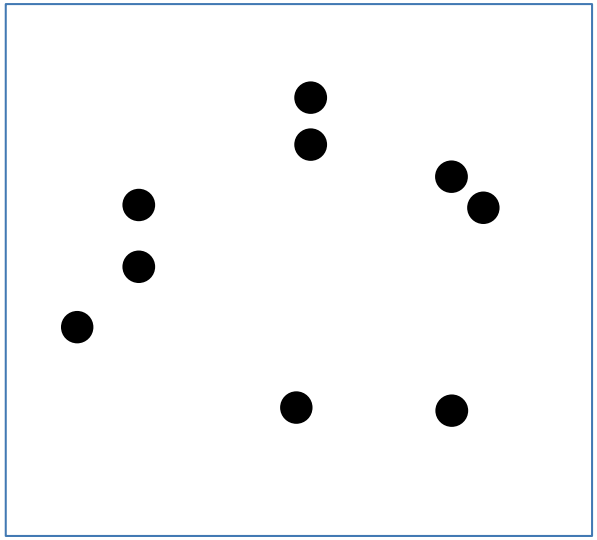


EVADoMe : Illustration de travaux

Algorithmes « *canopy* » : Principe

Partant d'un ensemble de points et une distance *d*'élire un certain nombre de représentants tels que :

- tout point est à distance au plus *d* d'un représentant
- tout représentant soit à distance au moins *d* des autres représentants

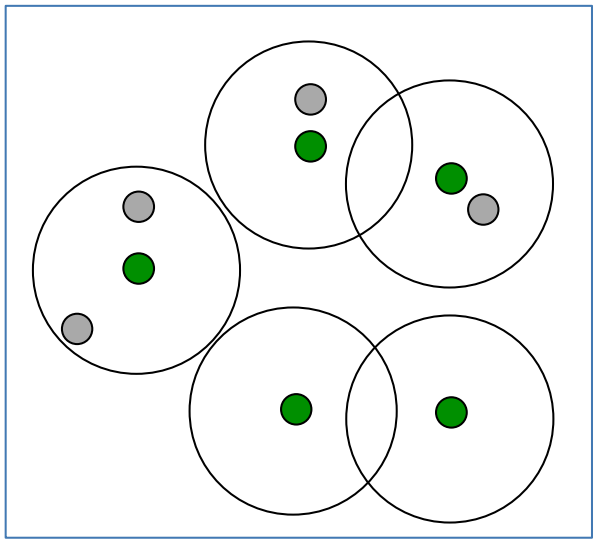


EVADoMe : Illustration de travaux

Algorithmes « *canopy* » : Principe

Partant d'un ensemble de points et une distance *d*'élire un certain nombre de représentants tels que :

- tout point est à distance au plus *d* d'un représentant
- tout représentant soit à distance au moins *d* des autres représentants

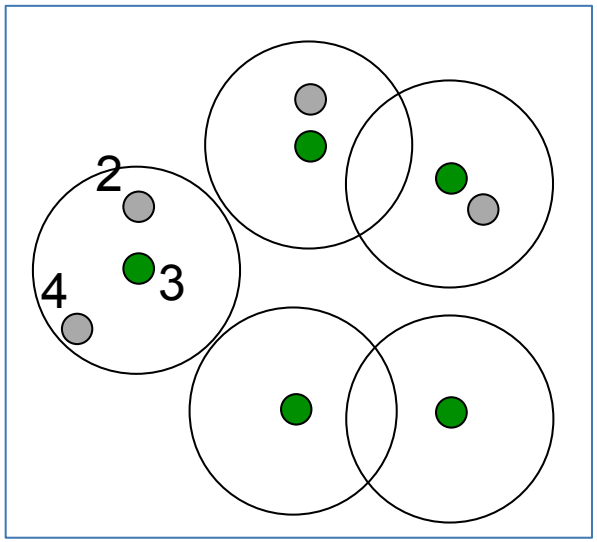


EVADoMe : Illustration de travaux

Algorithmes « *canopy* » : Principe

Partant d'un ensemble de points et une distance *d*'élire un certain nombre de représentants tels que :

- tout point est à distance au plus *d* d'un représentant
- tout représentant soit à distance au moins *d* des autres représentants

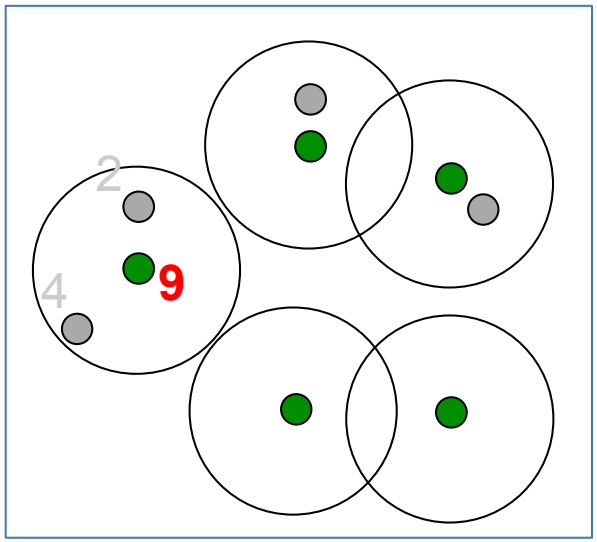


EVADoMe : Illustration de travaux

Algorithmes « *canopy* » : Principe

Partant d'un ensemble de points et une distance *d*'élire un certain nombre de représentants tels que :

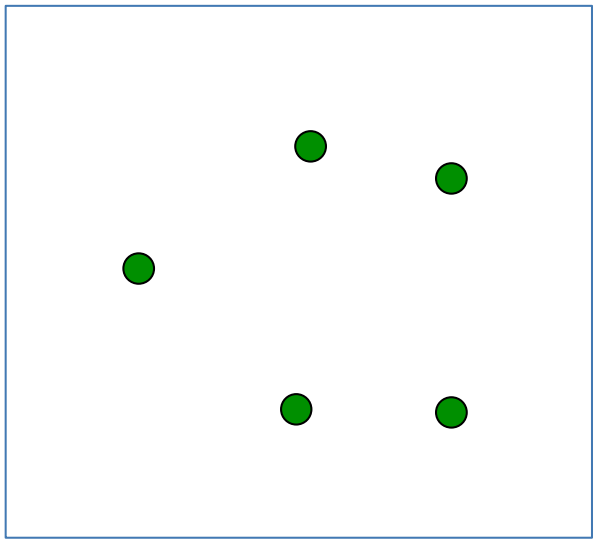
- tout point est à distance au plus *d* d'un représentant
- tout représentant soit à distance au moins *d* des autres représentants



EVADoMe : Illustration de travaux

Algorithmes « *canopy* » :
Partitionnement multi-niveaux

Appliquer itérativement l'algorithme canopy en :
• considérant les représentants du niveau précédent
• augmentant la distance *d*



EVADoMe : Illustration de travaux

Principe des algorithmes « *canopy* » :
Sur le cloud ?

Utilisation de 2 passes Map/Reduce

1ère passe :

- Division en bandes de l'espace (une bande par tâche)
- Calcul des représentants dans chaque bande

2ème passe :

- Gestion des conflits sur le « bord » des bandes

EVADoMe : Illustration de travaux

Principe des algorithmes « *canopy* » :
Résultat

- Partitionnement multi-niveaux
- À chaque point est associé les poids qu'il représente
- Garantie sur le nombre maximum de points par tuile

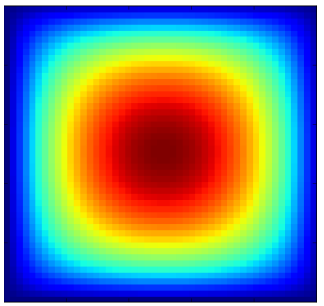
EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) : Principe

Considérant un ensemble de *n* points et une image de *p* pixels

- Calculer pour chaque pixel l'apport de chaque point (fonction de la distance / la pondération)
- Utiliser un noyau (généralement) gaussien

- Complexité en *O(n*p)*



EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) :
Heuristique en temps constant

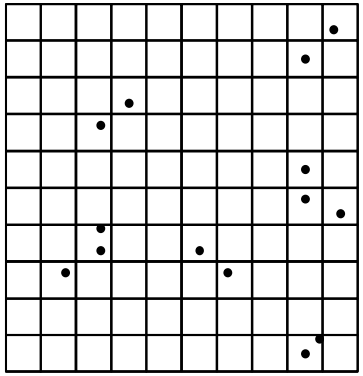
Principe :

Utiliser une implémentation sur la carte graphique et un découpage en tâches

- Chaque tâche calcule le KDE sur un sous ensemble de points et sur une image de résolution réduite
- Agrégation des images produites en une seule
- « Étirement » de l'image pour obtenir la résolution initiale

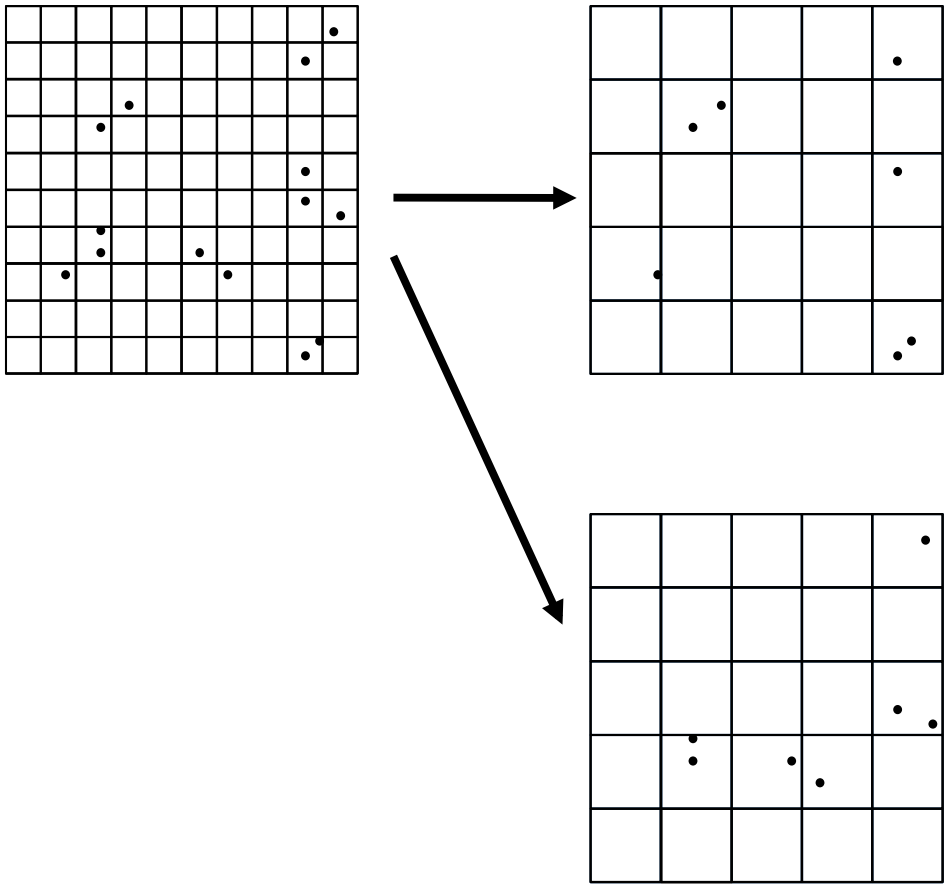
EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) : Heuristique en temps constant



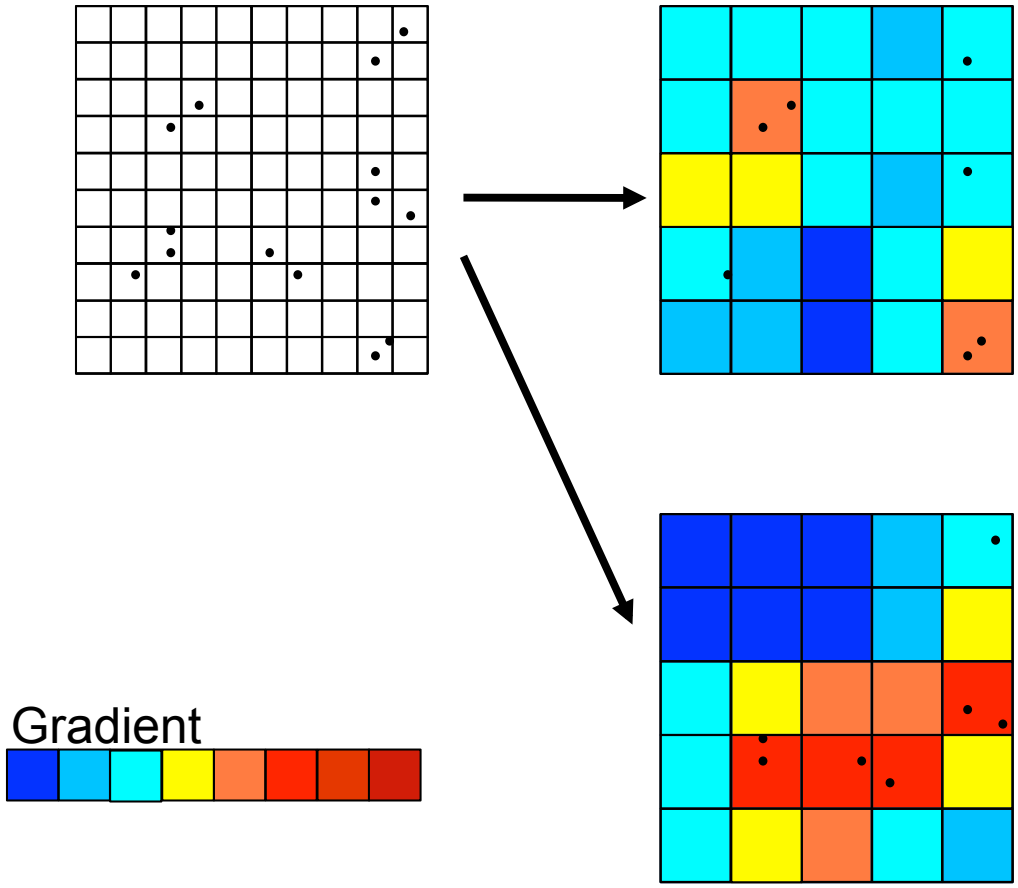
EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) : Heuristique en temps constant



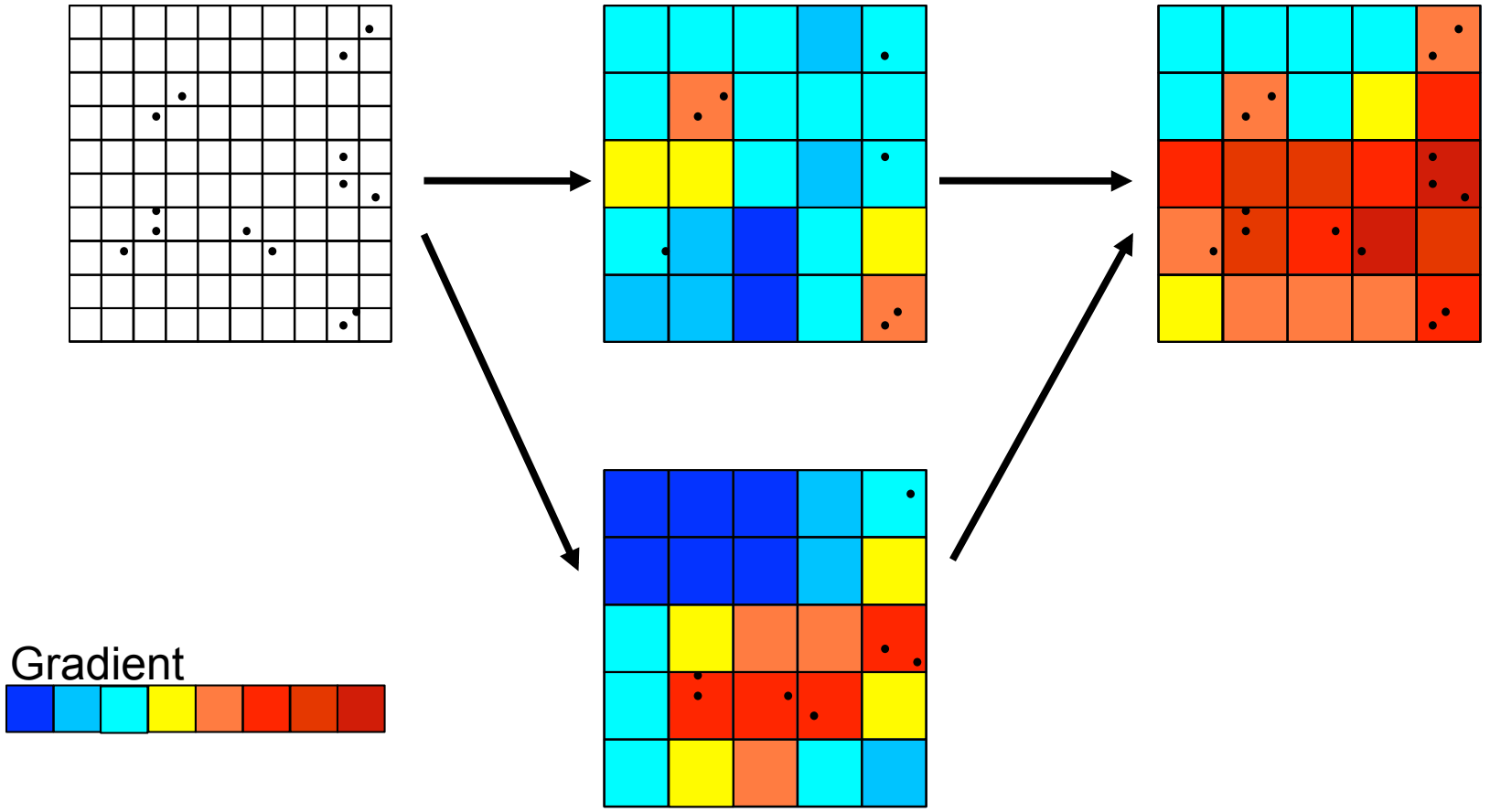
EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) :
Heuristique en temps constant



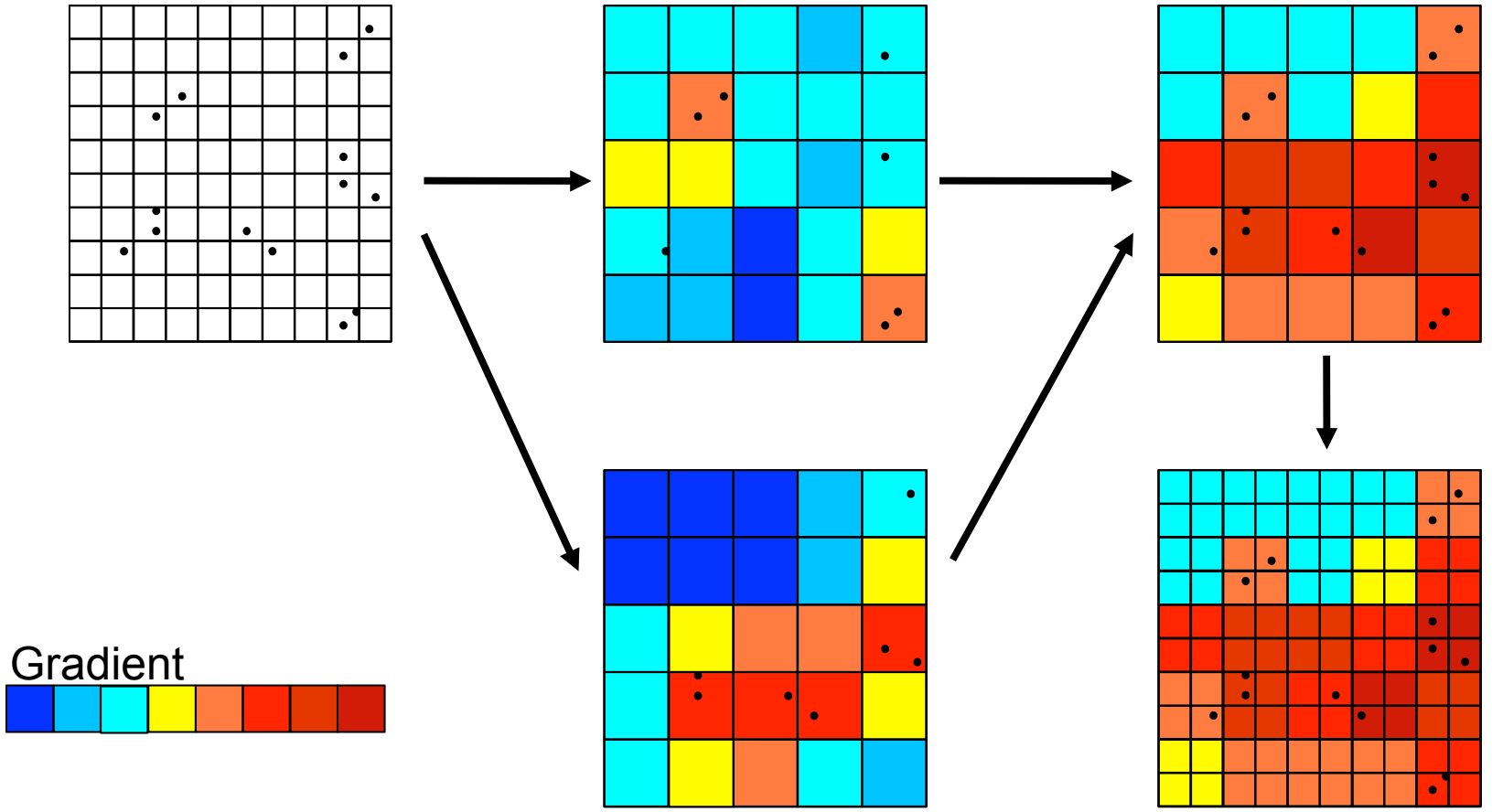
EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) :
Heuristique en temps constant



EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) :
Heuristique en temps constant



EVADoMe : Illustration de travaux

Estimation de densité de noyaux (*KDE*) :
Heuristique en temps constant

Principe :

Utiliser une implémentation sur la carte graphique et un découpage en tâches

- Chaque tâche calcule le KDE sur un sous ensemble de points et sur une image de résolution réduite
- Agrégation des images produites en une seule
- « Étirement » de l'image pour obtenir la résolution initiale

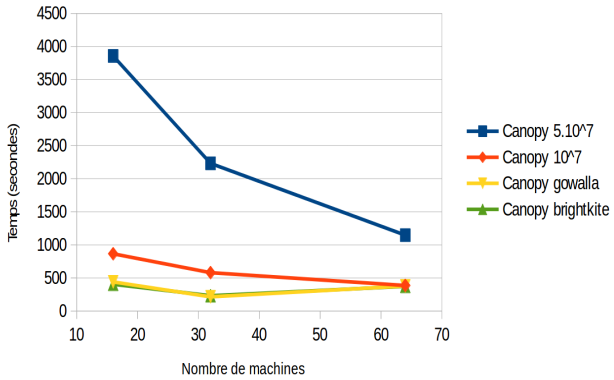
En temps constant (borné par une constante) mais cette constante dépend du nombre de pixels (fixe) et de la carte graphique

EVADoMe : Illustration de travaux

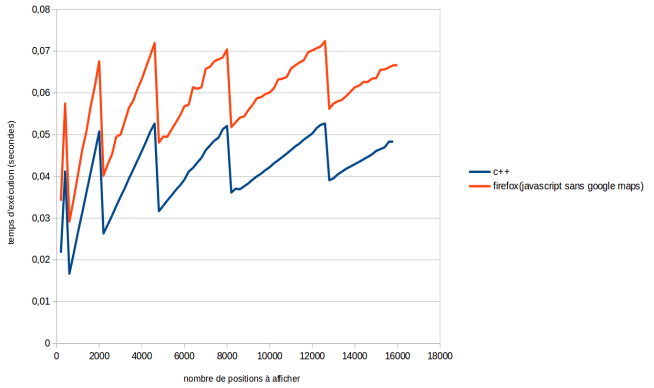
Évaluation

Temps de calcul

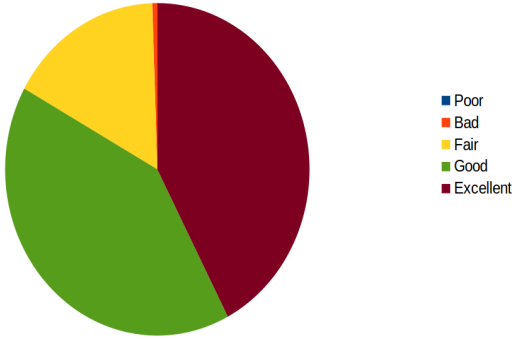
Partitionnement



Rendu (KDE)



Qualité



LaBRI