Adéquation Algorithme Architecture pour la reconstruction 3D en imagerie médicale TEP

Nicolas GAC

Thèse préparée à l'Institut Polytechnique de Grenoble (INPG)

Direction : *Michel DESVIGNES & Stéphane MANCINI*
Laboratoire : *Gipsa-lab* (Département Images et Signal), Grenoble

17 Juillet 2008

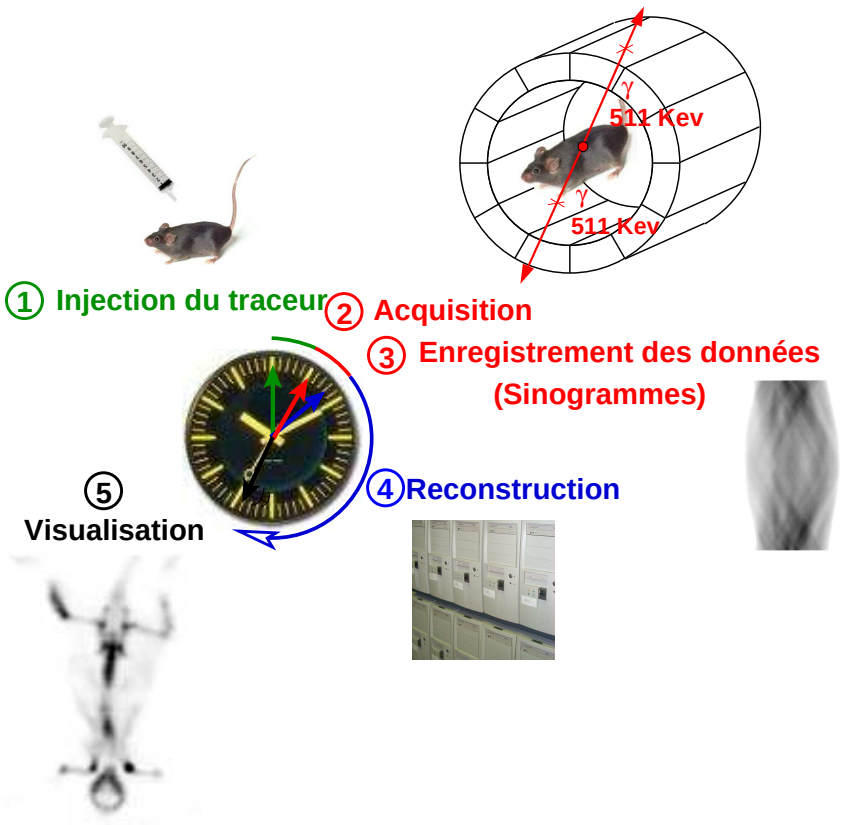
gipsa-lab

Grenoble INP

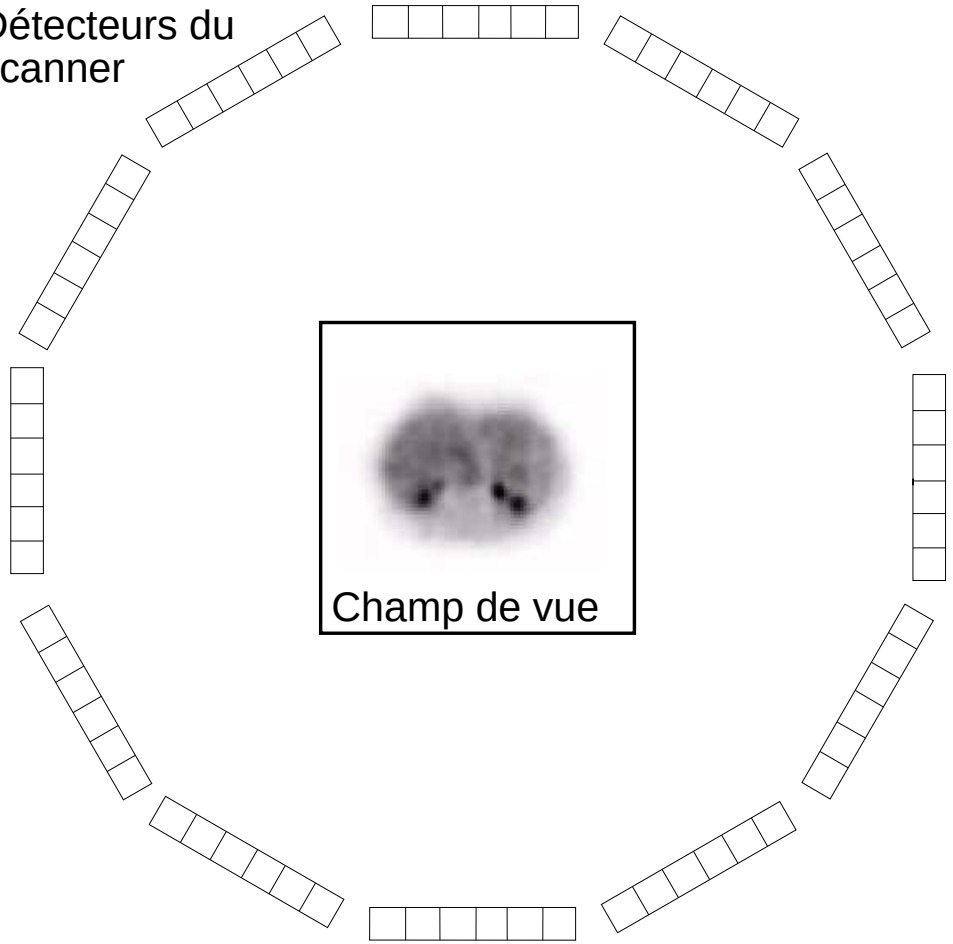
Projet ArchiTEP

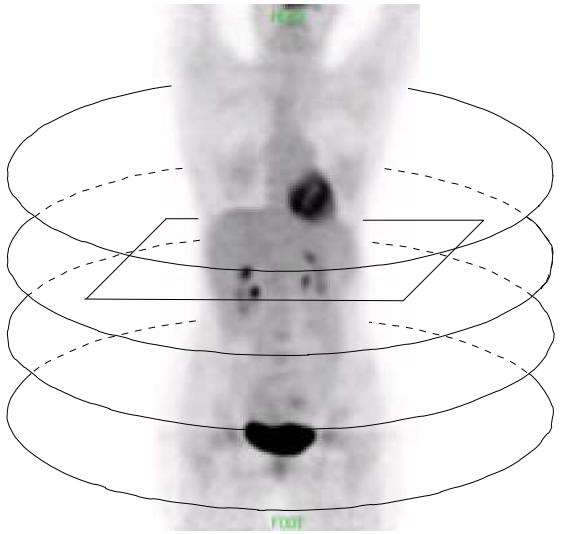
Tomographie à Emission de Positons (TEP)

- Imagerie fonctionnelle in vivo
- Temps de reconstruction importants



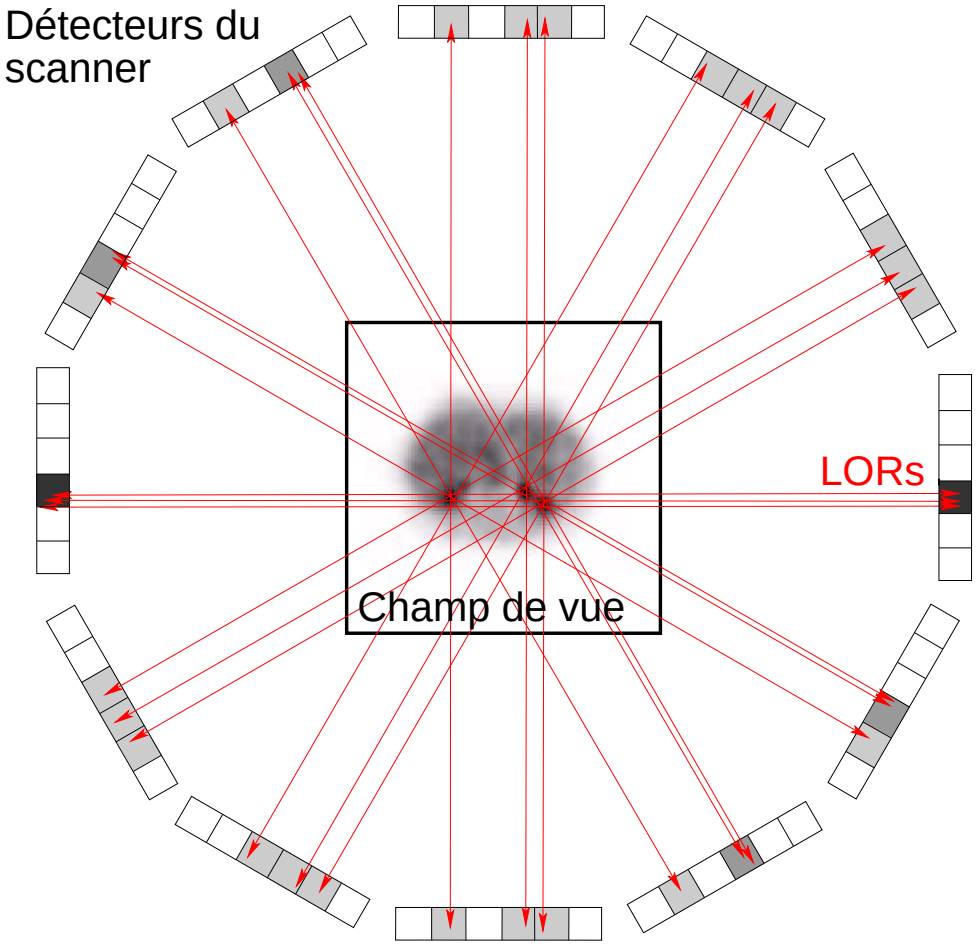
1 Injection du traceur

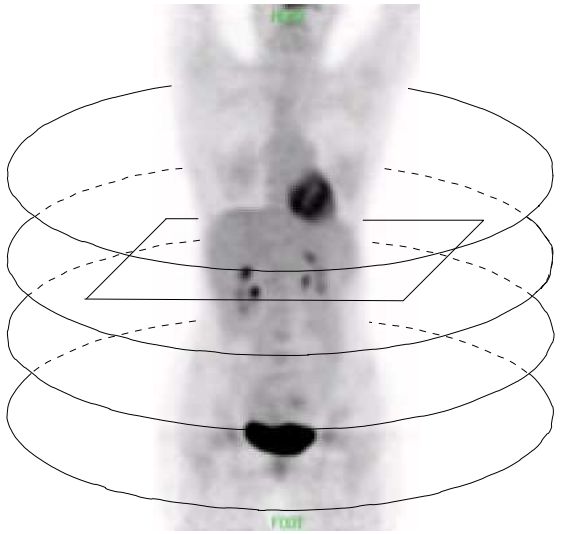




Volume

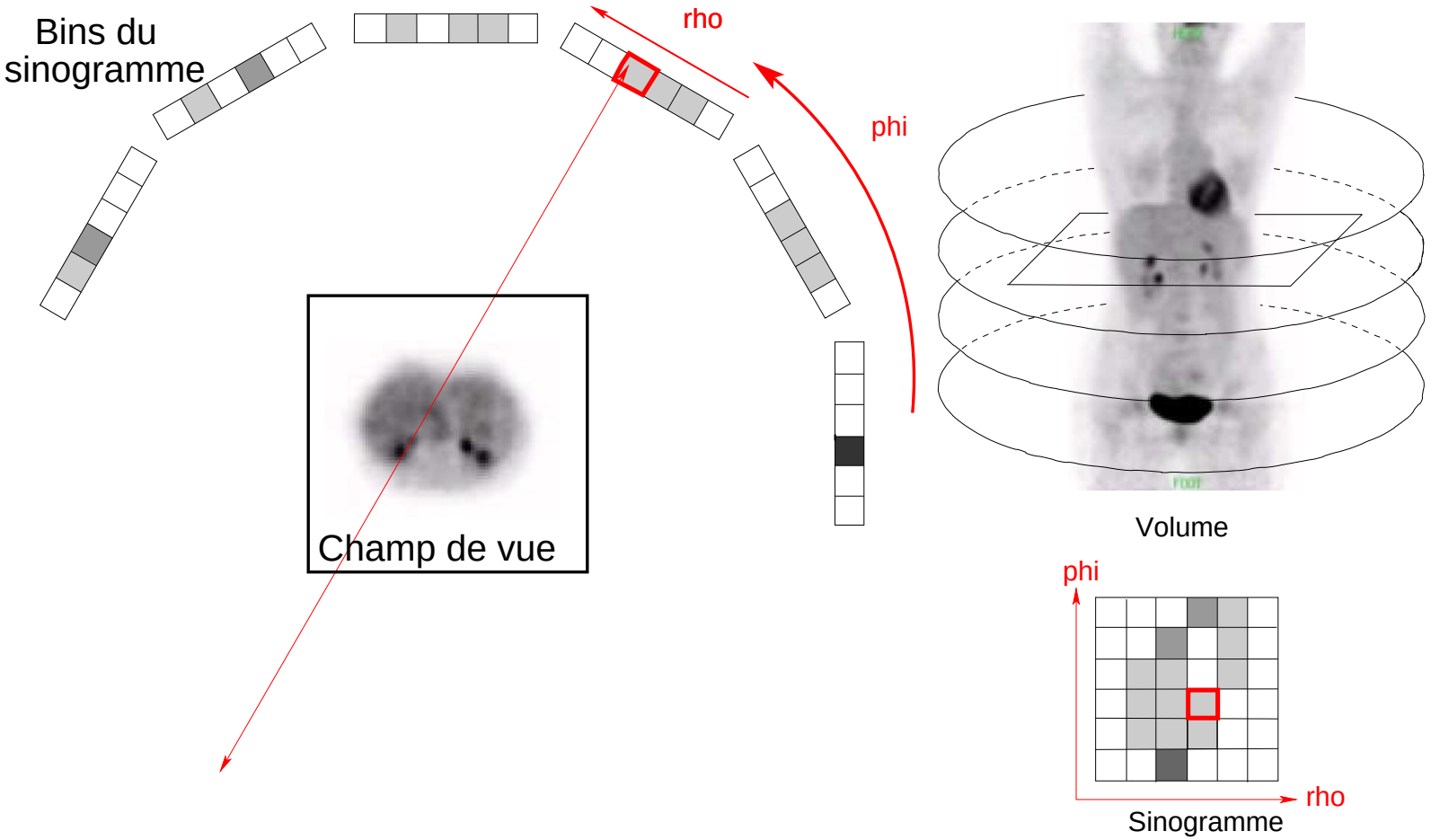
Détections des paires de photons émis



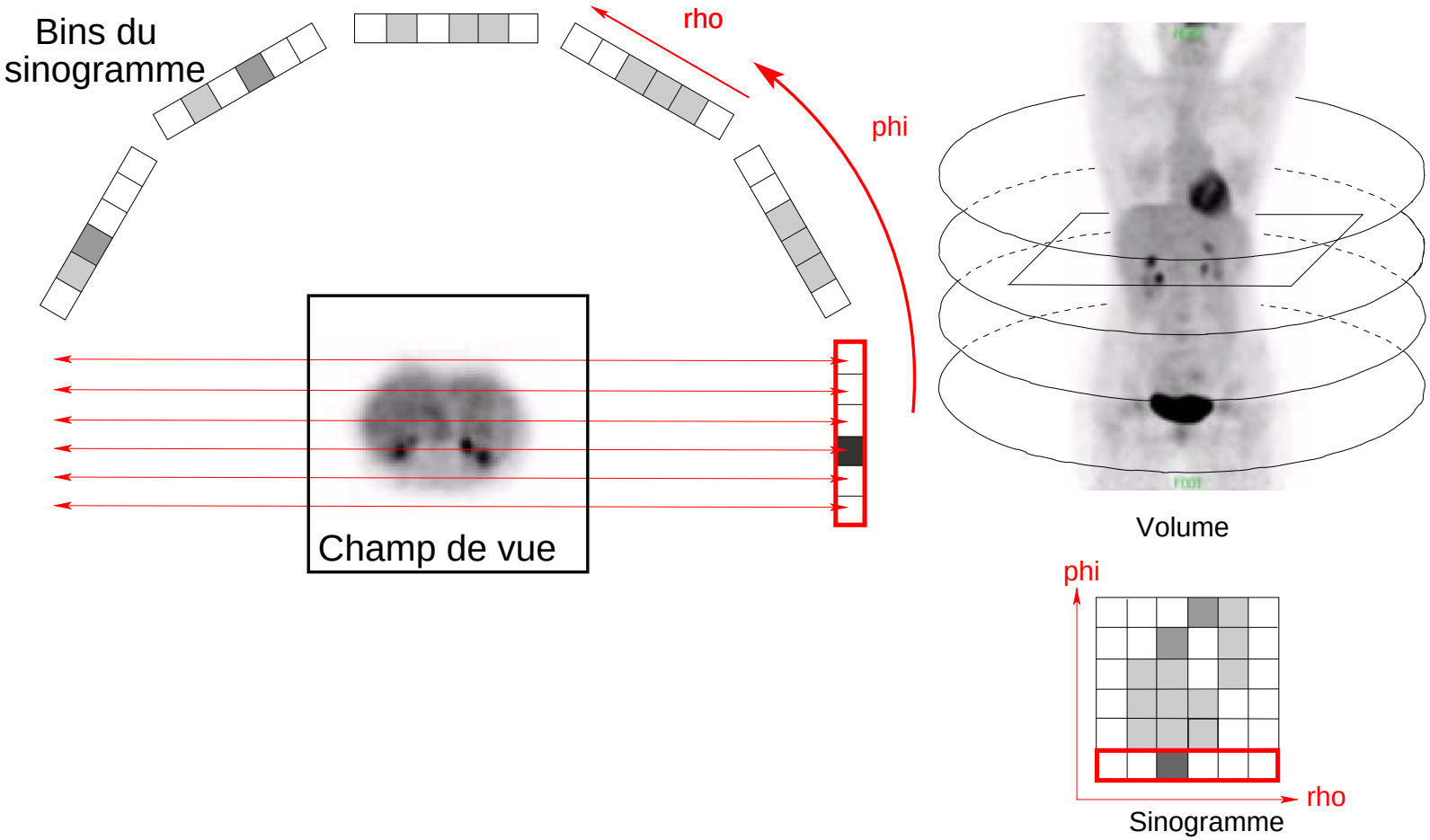


Volume

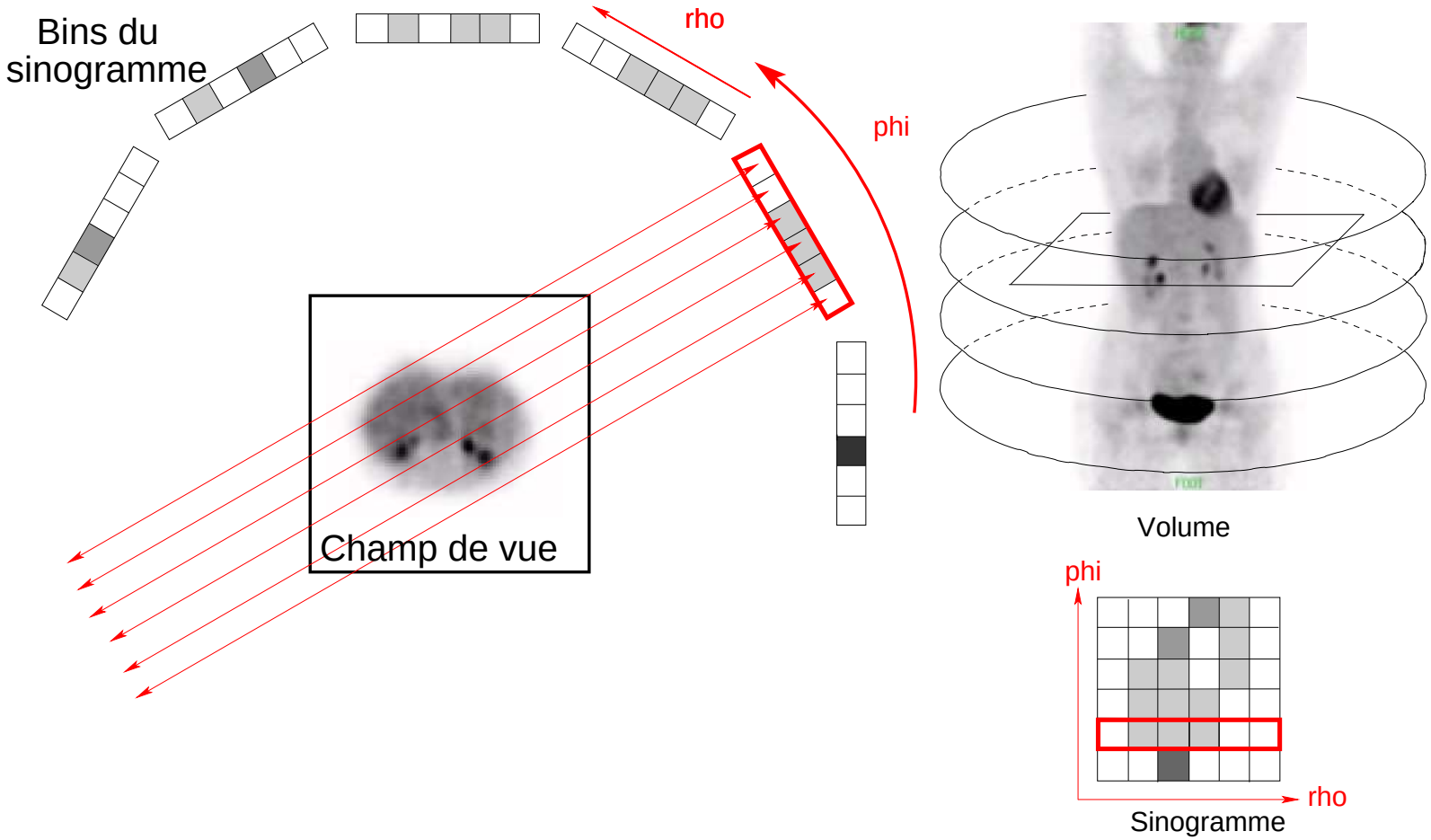
Stockage des données dans un sinogramme



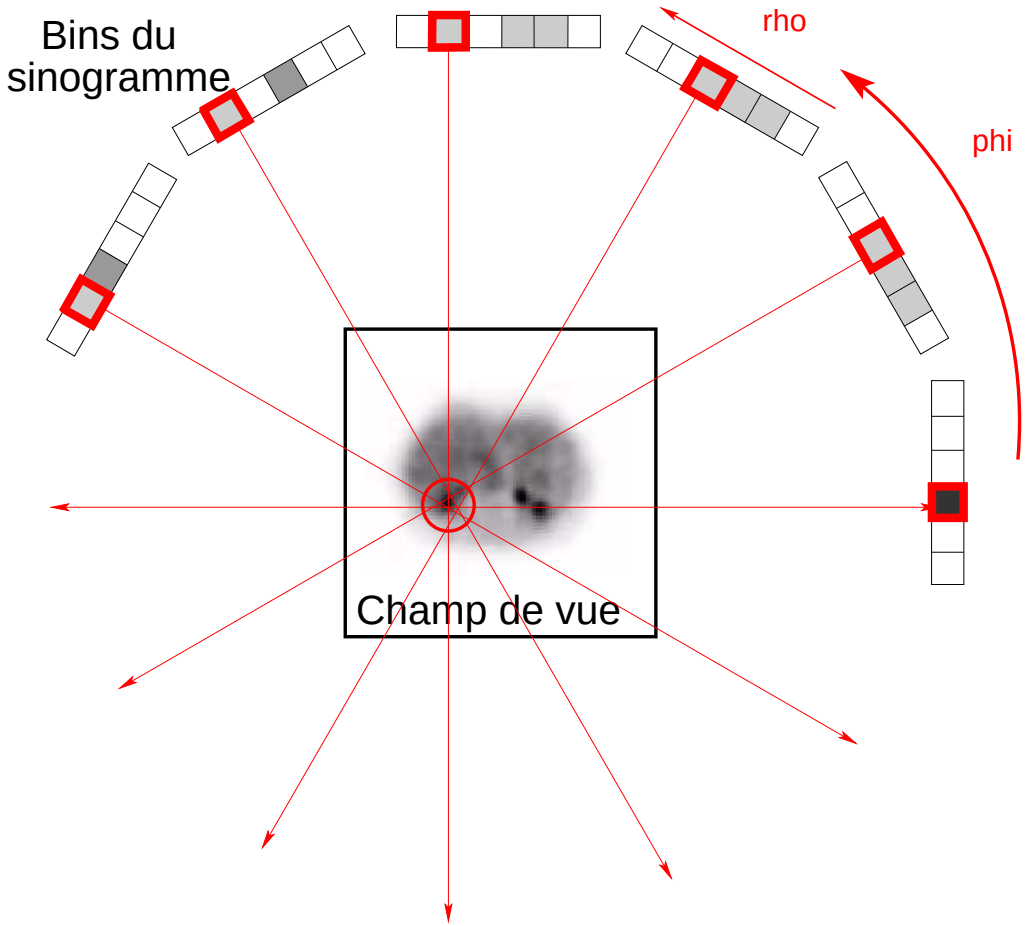
Stockage des données dans un sinogramme

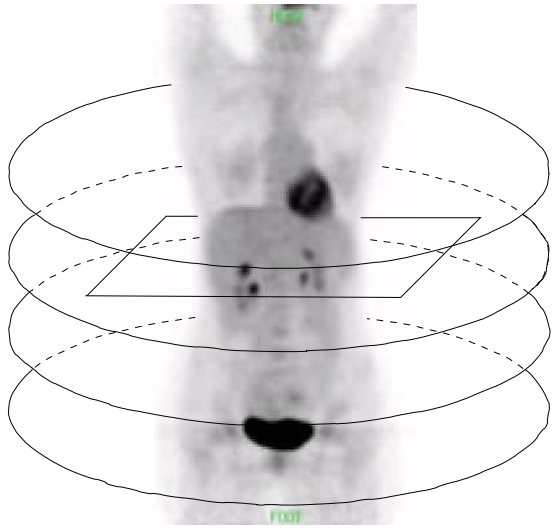


Stockage des données dans un sinogramme

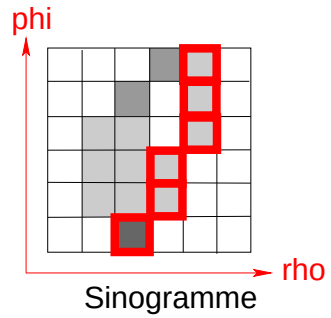


Stockage des données dans un sinogramme



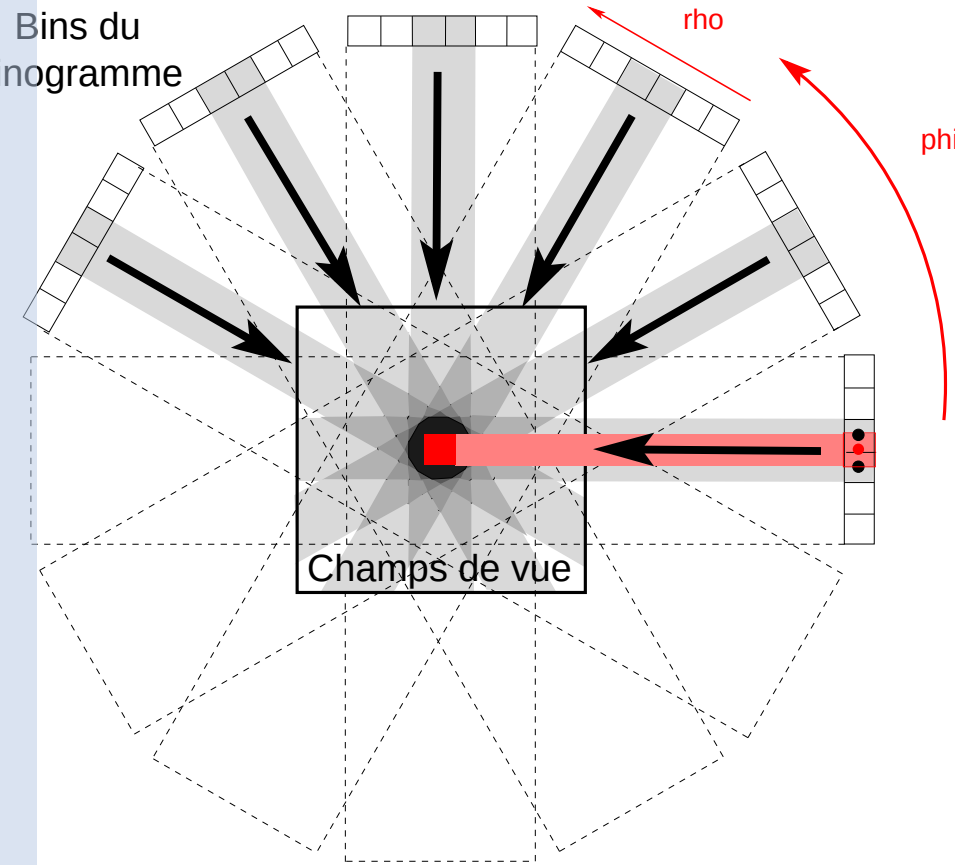


Volume



Sinogramme

Reconstruction du volume : rétroprojection 2D



Sommation des bins :

$$f^*(\mathtt{xn},\mathtt{yn}) = \sum_{\mathtt{phi}} \mathrm{bin}[\mathtt{phi},\mathtt{rho}(\mathtt{phi})]$$

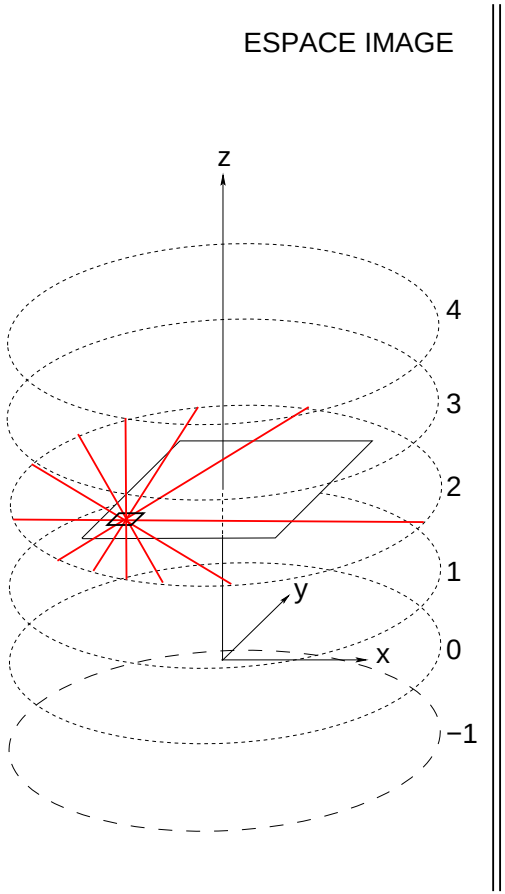
Calcul des coordonnées :

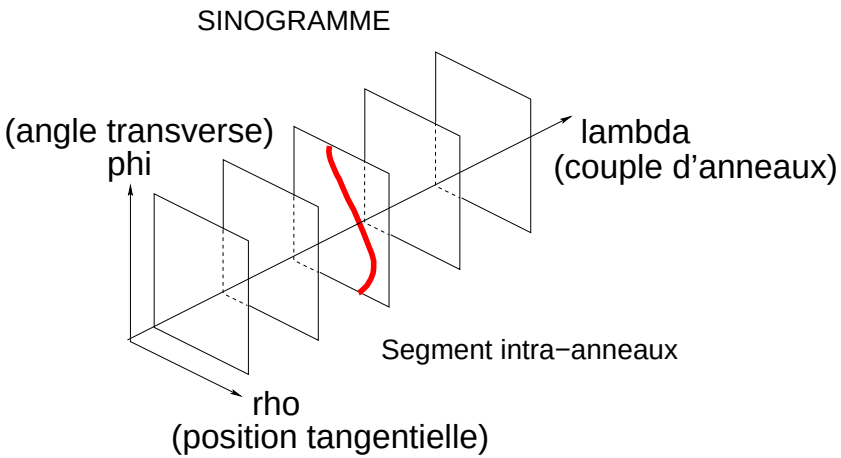
$$\begin{aligned} \mathtt{rho} &= \mathtt{xn}\cdot\cos(\phi) - \mathtt{yn}\cdot\sin(\phi) \\ &= \mathtt{rho_e} + \epsilon_\rho \end{aligned}$$

Interpolation linéaire :

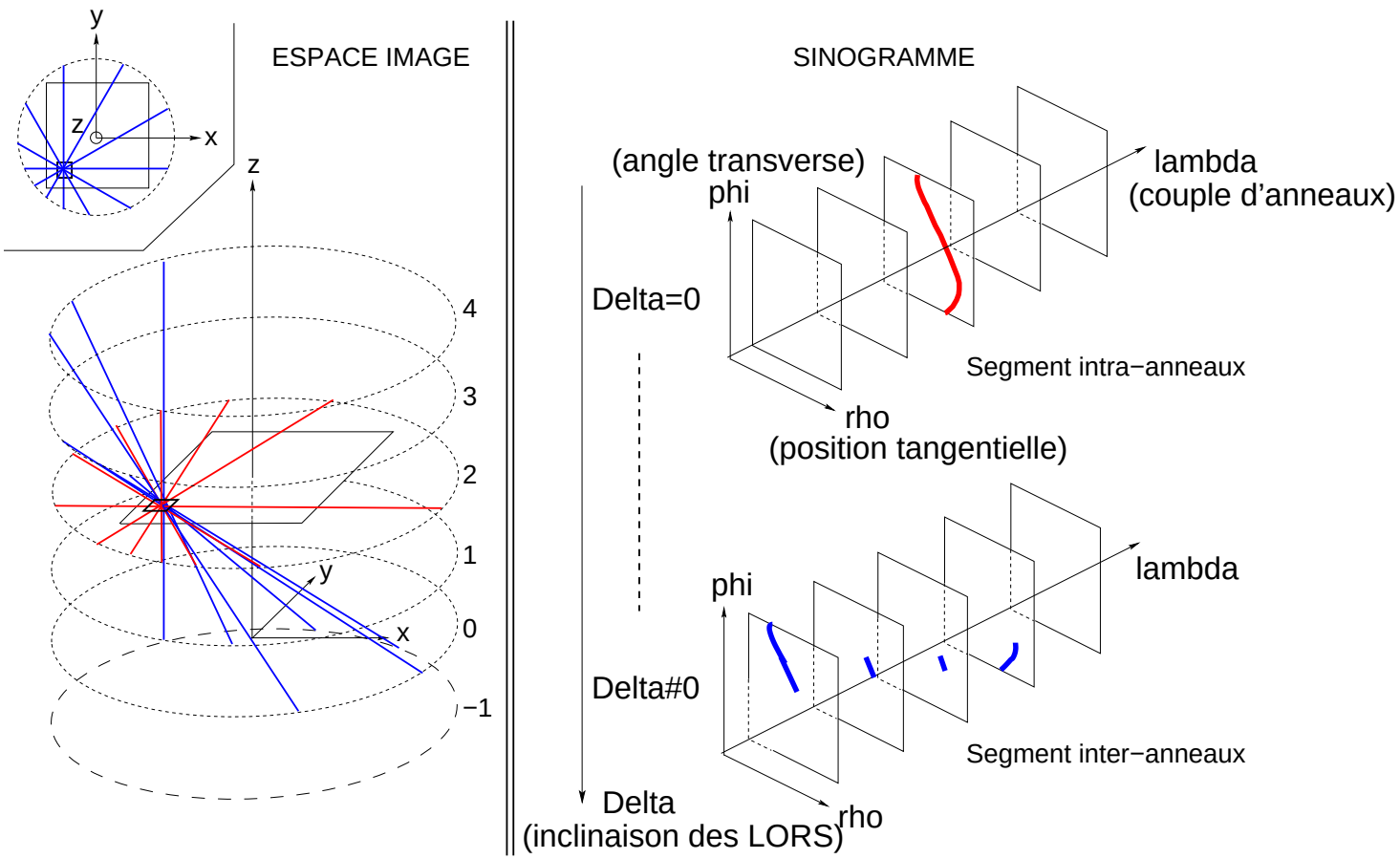
$$\begin{aligned} \mathrm{bin} = \quad & (1-\epsilon_\rho)\cdot \mathrm{bin}(\mathtt{phi},\mathtt{rho_e}) \\ & + \\ & \epsilon_\rho \cdot \mathrm{bin}(\mathtt{phi},\mathtt{rho_e}+1) \end{aligned}$$

Acquisition 2D/3D





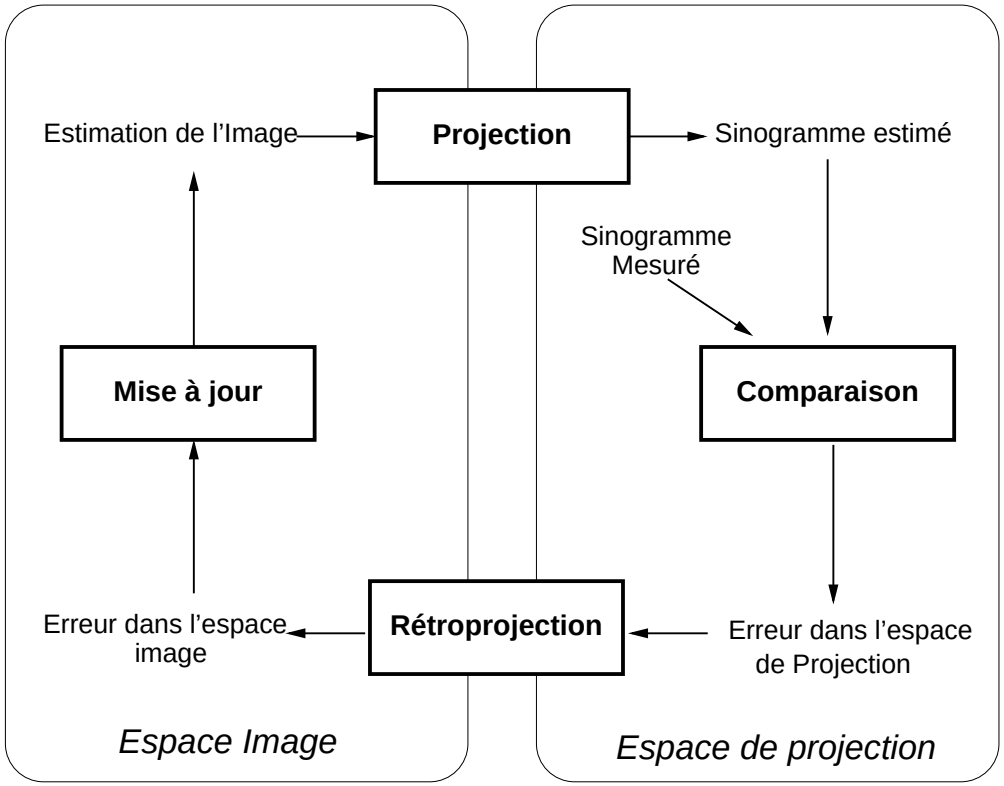
Acquisition 2D/3D



- Acquisition 3D ➲ Meilleure qualité de reconstruction

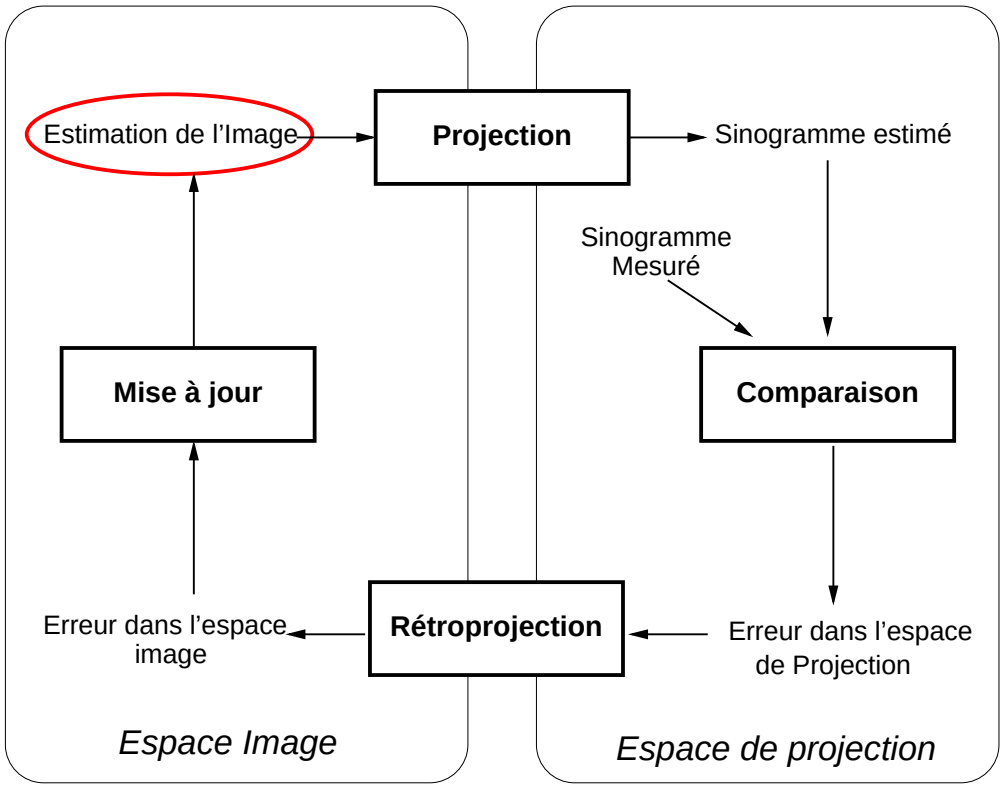
Méthodes itératives

- Meilleure qualité de reconstruction
- Temps de reconstruction plus important



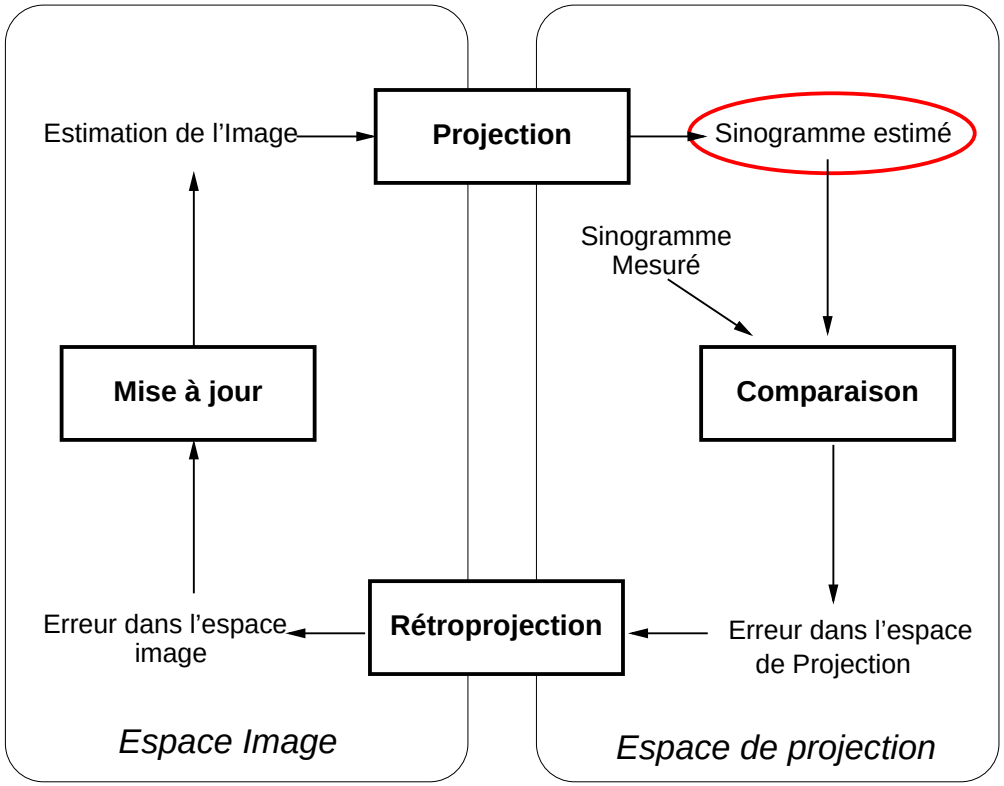
Méthodes itératives

- Meilleure qualité de reconstruction
- Temps de reconstruction plus important



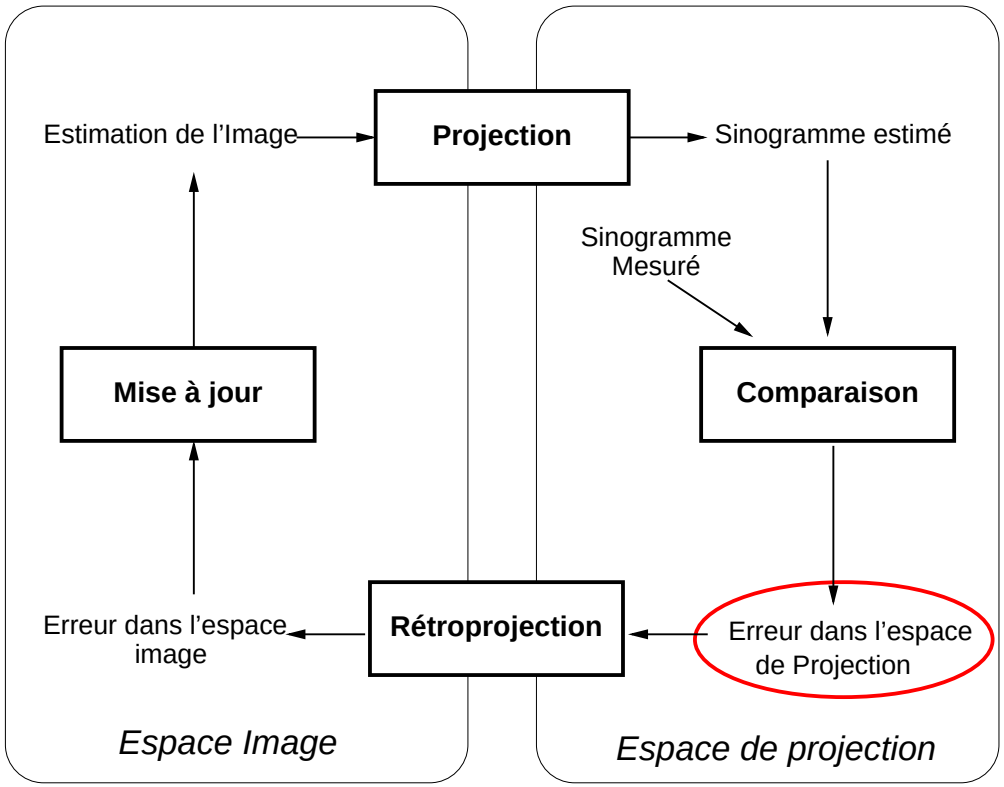
Méthodes itératives

- Meilleure qualité de reconstruction
- Temps de reconstruction plus important



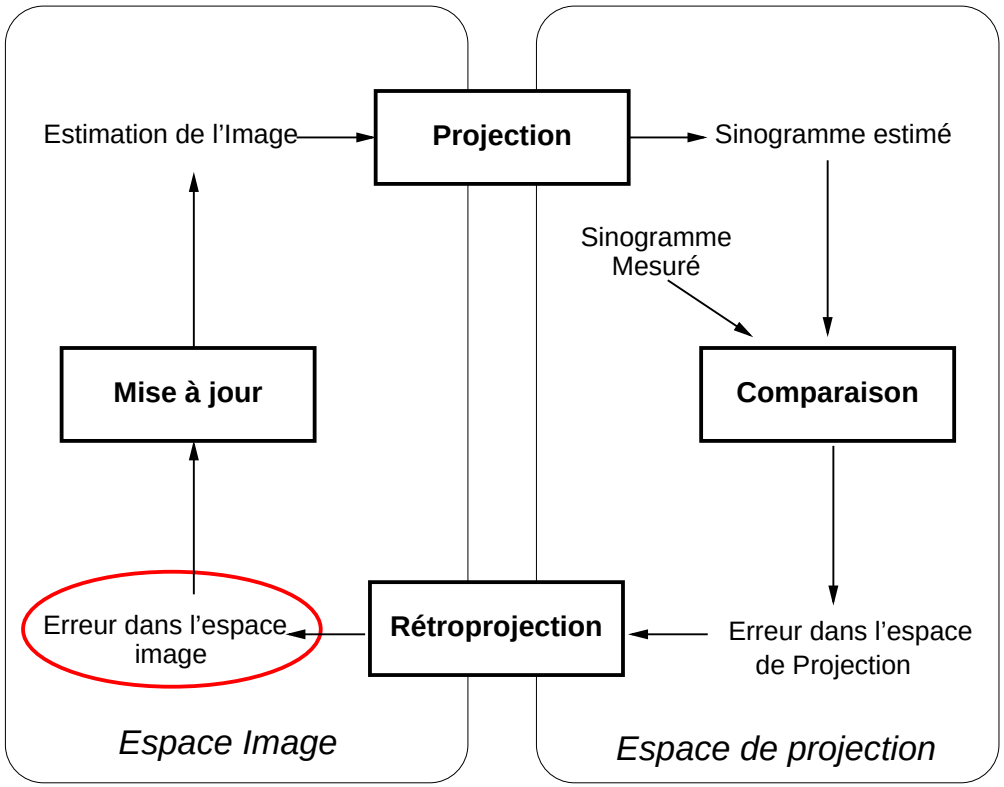
Méthodes itératives

- Meilleure qualité de reconstruction
- Temps de reconstruction plus important



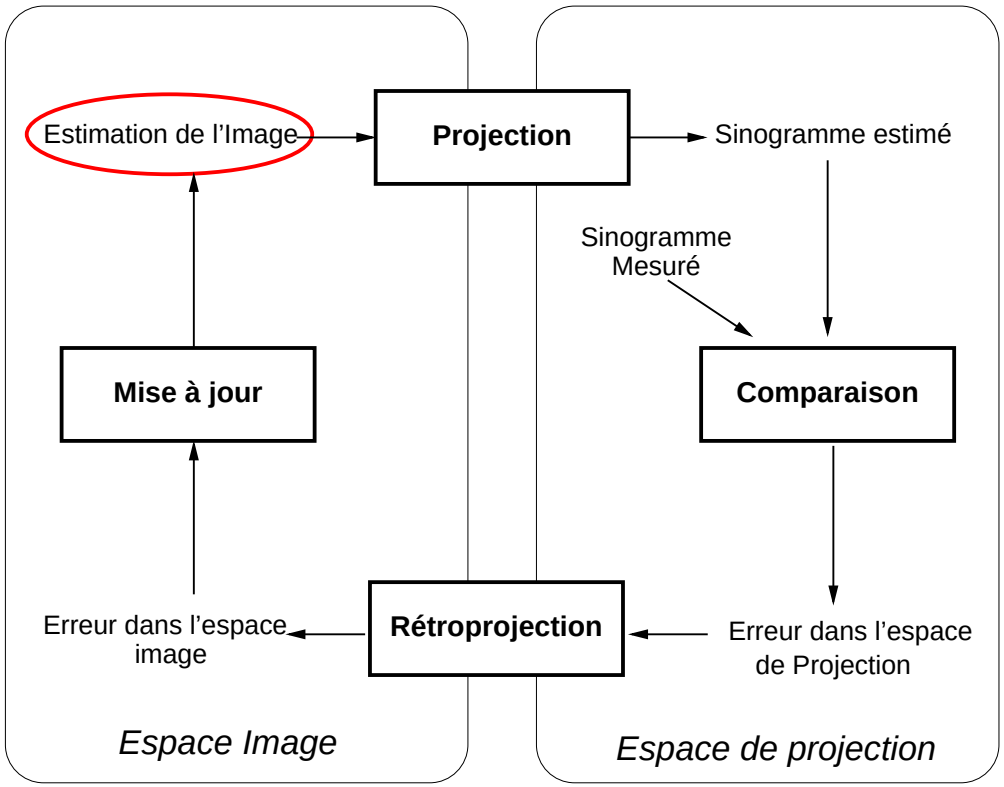
Méthodes itératives

- Meilleure qualité de reconstruction
- Temps de reconstruction plus important



Méthodes itératives

- Meilleure qualité de reconstruction
- Temps de reconstruction plus important



Nécessité d'accélérer la reconstruction

Un problème de plus en plus complexe

- Amélioration de la résolution spatiale des scanners
  - ➲ *Volume de* $\simeq 10^6$ *voxels*
  - ➲ *Sinogramme de* $\simeq 500 \cdot 10^6$ *bins pour le scanner HRRT*
  - ➲ $\simeq$ *1 000 mises à jour par voxel*
- Utilisation de méthodes itératives
  - ➲ *30 à 40 itérations nécéssaires en EM*
- Reconstruction 4D en TEP dynamique
  - ➲ *30/60 frames à reconstruire*

Temps de reconstruction insuffisant sur PCs

- 16 heures de calcul en OSEM sur un scanner HRRT
- Retard technologique de 10/15 ans par rapport aux scanners

Accélération matérielle

❶ Parallélisation sur machines multi-processeurs

- Efficace sur machine à mémoire distribuée
- Inefficace sur machine à mémoire centralisée
  ➲ L'accès à la mémoire est un goulot d'étranglement

❷ Noeuds de calcul performants

- Multi-Processor System on Chip (MPSoC) (*Cell*, *GP-GPU*)
- System on Programmable Chip (SoPC)
  ➲ Nécessité d'une stratégie efficace d'accès mémoire

Accélération matérielle

❶ Parallélisation sur machines multi-processeurs

- Efficace sur machine à mémoire distribuée
- Inefficace sur machine à mémoire centralisée
  ➲ L'accès à la mémoire est un goulot d'étranglement

❷ Noeuds de calcul performants

- Multi-Processor System on Chip (MPSoC) (*Cell*, *GP-GPU*)
- System on Programmable Chip (SoPC)
  ➲ Nécessité d'une stratégie efficace d'accès mémoire

Algorithme de rétroprojection 3D

for $(\mathtt{xn}, \mathtt{yn}, \mathtt{zn})$ in volume **do**
  for $\mathtt{delta} = 0$ to $\mathtt{delta}_{\mathtt{max}} - 1$ **do**
    for $\mathtt{phi} = 0$ to $\mathtt{phi}_{\mathtt{max}} - 1$ **do**
      // CALCUL DES COORDONNEES
      $\mathtt{rho}(\mathtt{phi}) = \mathtt{xn} \cdot \cos\phi + \mathtt{yn} \cdot \sin\phi$
      $\mathtt{lambda}(\mathtt{phi}, \mathtt{delta}) = \ldots$
      // INTERPOLATION BI-LINEAIRE
      $\mathrm{bin}_{interp} = C_{00} \cdot \mathrm{bin}_{00} + C_{01} \cdot \mathrm{bin}_{01} \ \ldots$
      // ACCUMULATION
      $f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) = f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) + \mathrm{bin}_{interp} \cdot J_\Delta$
    end for
  end for
end for

Algorithme de rétroprojection 3D

for $(\mathtt{xn}, \mathtt{yn}, \mathtt{zn})$ in volume **do**
  for $\mathtt{delta} = 0$ to $\mathtt{delta}_{\mathtt{max}} - 1$ **do**
    for $\mathtt{phi} = 0$ to $\mathtt{phi}_{\mathtt{max}} - 1$ **do**
      // CALCUL DES COORDONNEES
      $\mathtt{rho}(\mathtt{phi}) = \mathtt{xn} \cdot \cos\phi + \mathtt{yn} \cdot \sin\phi$
      $\mathtt{lambda}(\mathtt{phi}, \mathtt{delta}) = \ldots$
      // INTERPOLATION BI-LINEAIRE
      $\mathrm{bin}_{interp} = C_{00} \cdot \mathrm{bin}_{00} + C_{01} \cdot \mathrm{bin}_{01} \ \ldots$
      // ACCUMULATION
      $f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) = f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) + \mathrm{bin}_{interp} \cdot J_\Delta$
    end for
  end for
end for

Algorithme de rétroprojection 3D

for $(\mathtt{xn}, \mathtt{yn}, \mathtt{zn})$ in volume **do**
  for $\mathtt{delta} = 0$ to $\mathtt{delta}_{\mathtt{max}} - 1$ **do**
    for $\mathtt{phi} = 0$ to $\mathtt{phi}_{\mathtt{max}} - 1$ **do**
      // CALCUL DES COORDONNEES
      $\mathtt{rho}(\mathtt{phi}) = \mathtt{xn} \cdot \cos\phi + \mathtt{yn} \cdot \sin\phi$
      $\mathtt{lambda}(\mathtt{phi}, \mathtt{delta}) = \ldots$
      // INTERPOLATION BI-LINEAIRE
      $\mathrm{bin}_{interp} = C_{00} \cdot \mathrm{bin}_{00} + C_{01} \cdot \mathrm{bin}_{01} \ \ldots$
      // ACCUMULATION
      $f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) = f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) + \mathrm{bin}_{interp} \cdot J_\Delta$
    end for
  end for
end for

Algorithme de rétroprojection 3D

for $(\mathtt{xn}, \mathtt{yn}, \mathtt{zn})$ in volume **do**
  for $\mathtt{delta} = 0$ to $\mathtt{delta}_{\mathtt{max}} - 1$ **do**
    for $\mathtt{phi} = 0$ to $\mathtt{phi}_{\mathtt{max}} - 1$ **do**
      // CALCUL DES COORDONNEES
      $\mathtt{rho}(\mathtt{phi}) = \mathtt{xn} \cdot \cos\phi + \mathtt{yn} \cdot \sin\phi$
      $\mathtt{lambda}(\mathtt{phi}, \mathtt{delta}) = \ldots$
      // INTERPOLATION BI-LINEAIRE
      $\mathrm{bin}_{interp} = C_{00} \cdot \mathrm{bin}_{00} + C_{01} \cdot \mathrm{bin}_{01} \ \ldots$
      // ACCUMULATION
      $f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) = f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) + \mathrm{bin}_{interp} \cdot J_\Delta$
    end for
  end for
end for

Algorithme de rétroprojection 3D

Séquence de calcul

- Boucles imbriqués sans dépendance de données
  ➲ *Algorithme massivement parallèle*
- Séquence simple de calcul pour la mise à jour d'un voxel
  ➲ *Architecture en pipeline*

Accès mémoire

- Sinogramme de taille importante (10 Mo à 1 Go)
  ➲ *Stockage du sinogramme en SDRAM*
- Accès à 4 *bins* par mise à jour de voxel
  ➲ *Nécessité de masquer le temps d'accès en SDRAM*

Stratégie d'accès mémoire

❶ Réutilisation des données

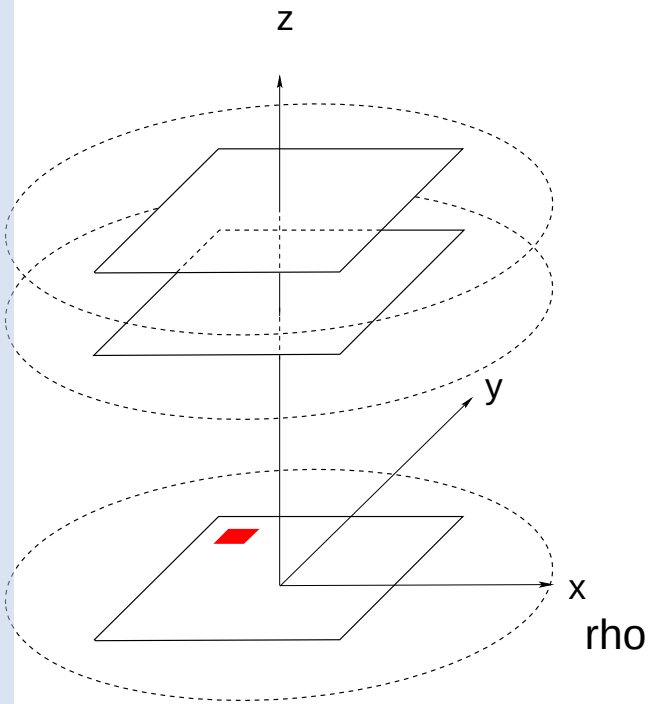
- Algorithme ➲ *localité temporelle*

❷ Mécanisme de préchargement mémoire

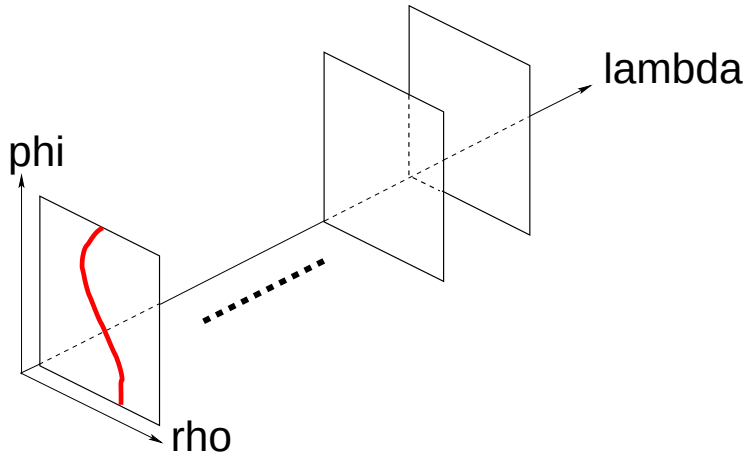
- "ad hoc" (*double buffering*)
  ➲ *Ressources de calcul supplémentaires*
  ➲ *Mécanisme de prédiction figé*
- générique
  Algorithme ➲ *localité spatio-temporelle*
  Architecture ➲ *cache mémoire "intelligent"*

Taux de réutilisation des données

ESPACE IMAGE

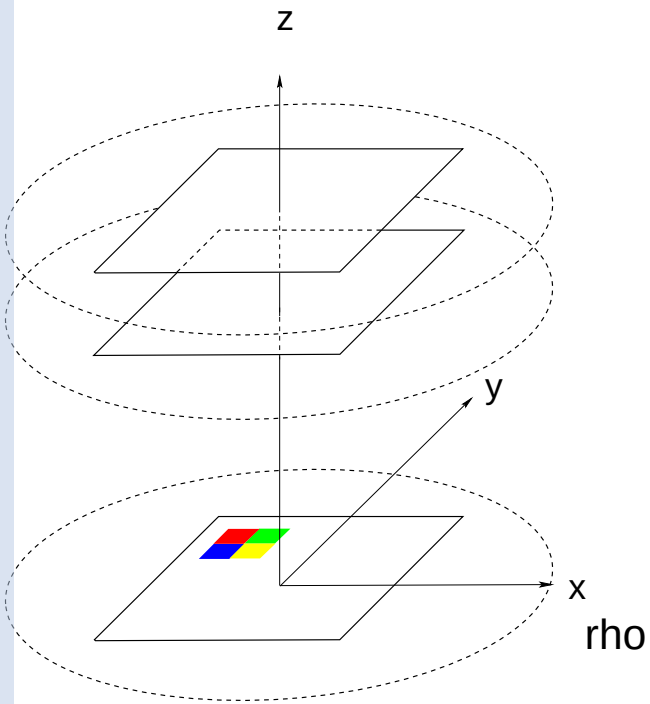


SINOGRAMME

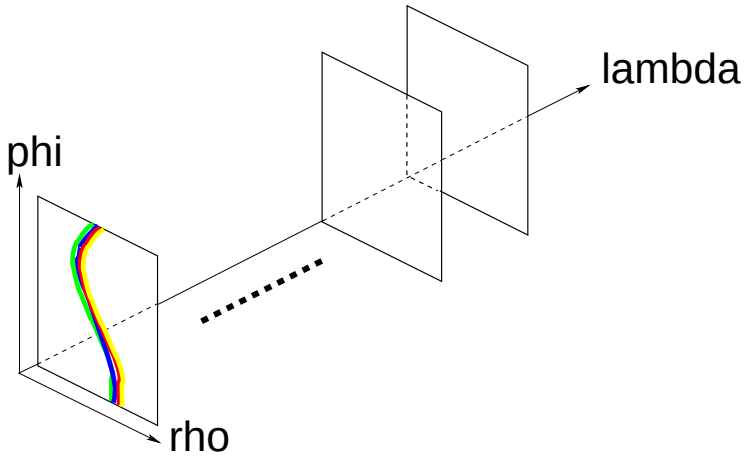


Taux de réutilisation des données

ESPACE IMAGE

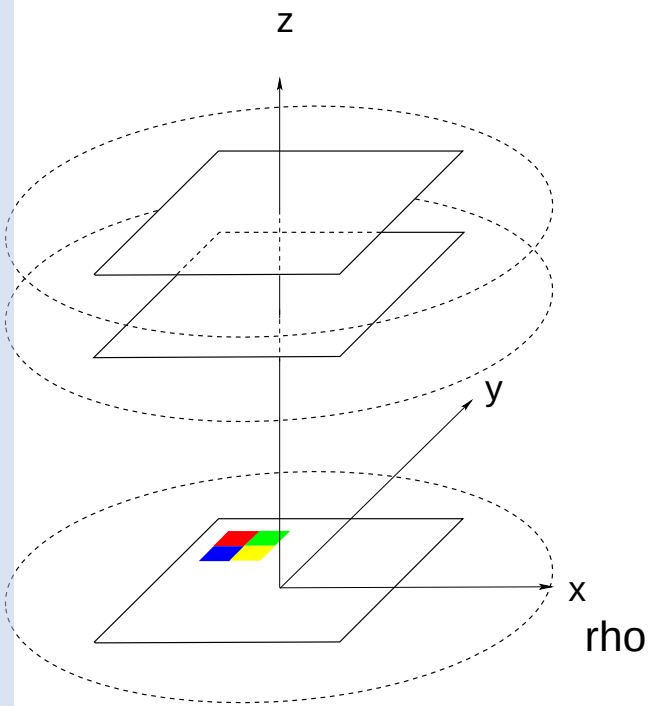


SINOGRAMME

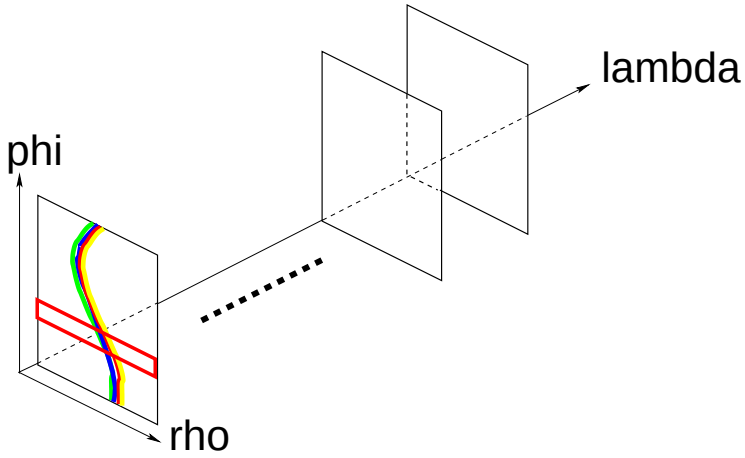


Taux de réutilisation des données

ESPACE IMAGE



SINOGRAMME



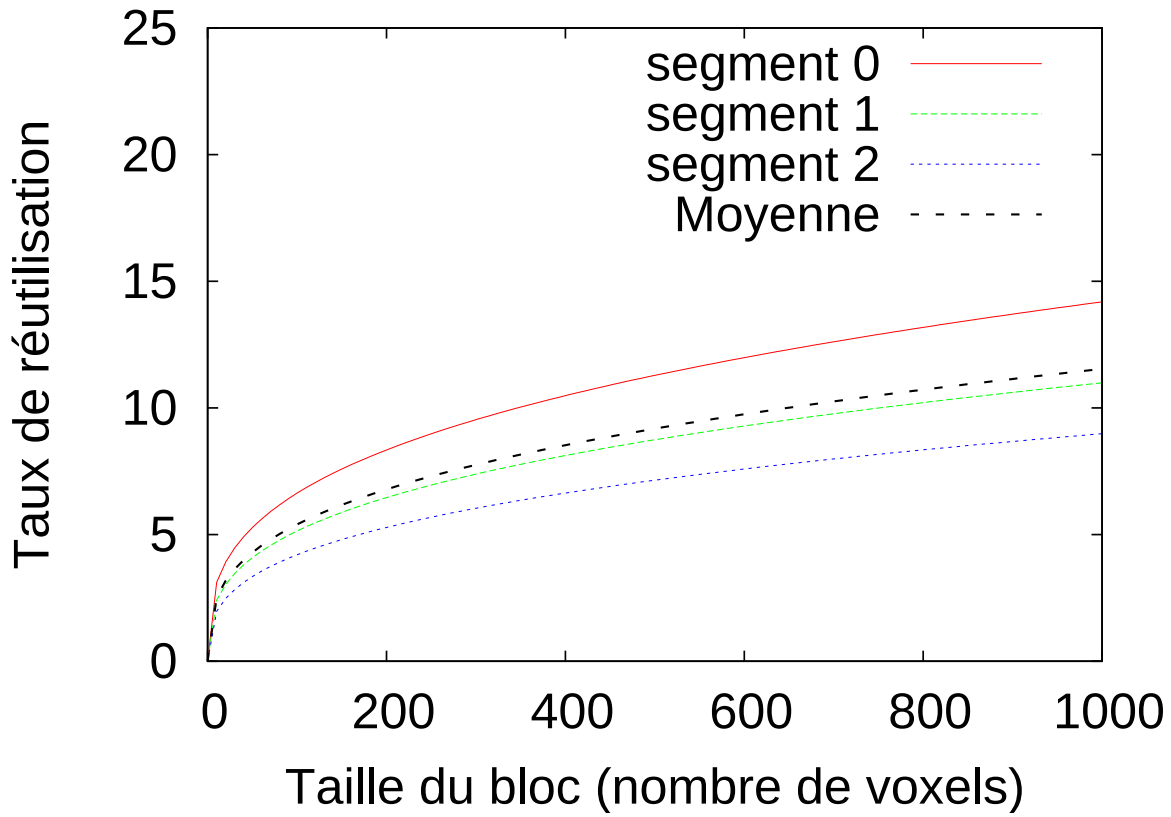
Reconstruction par blocs de voxels

for Bloc in Volume **do**
  for $\mathtt{delta} = 0$ to $\mathtt{delta}_{\mathtt{max}} - 1$ **do**
    for $\mathtt{phi} = 0$ to $\mathtt{phi}_{\mathtt{max}} - 1$ **do**
      for $(\mathtt{xn}, \mathtt{yn}, \mathtt{zn})$ in Bloc **do**
        // CALCUL DES COORDONNEES
        $\mathtt{rho}(\mathtt{phi}) = \mathtt{xn} \cdot \cos\phi + \mathtt{yn} \cdot \sin\phi$
        $\mathtt{lambda}(\mathtt{phi}, \mathtt{delta}) = ...$
        // INTERPOLATION BILINEAIRE
        $\mathrm{bin}_{interp} = C_{00} \cdot \mathrm{bin}_{00} + C_{01} \cdot \mathrm{bin}_{01}$ ...
        // ACCUMULATION
        $f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) = f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) + \mathrm{bin}_{interp} \cdot J_\Delta$
      end for
    end for
  end for
end for

Reconstruction par blocs de voxels

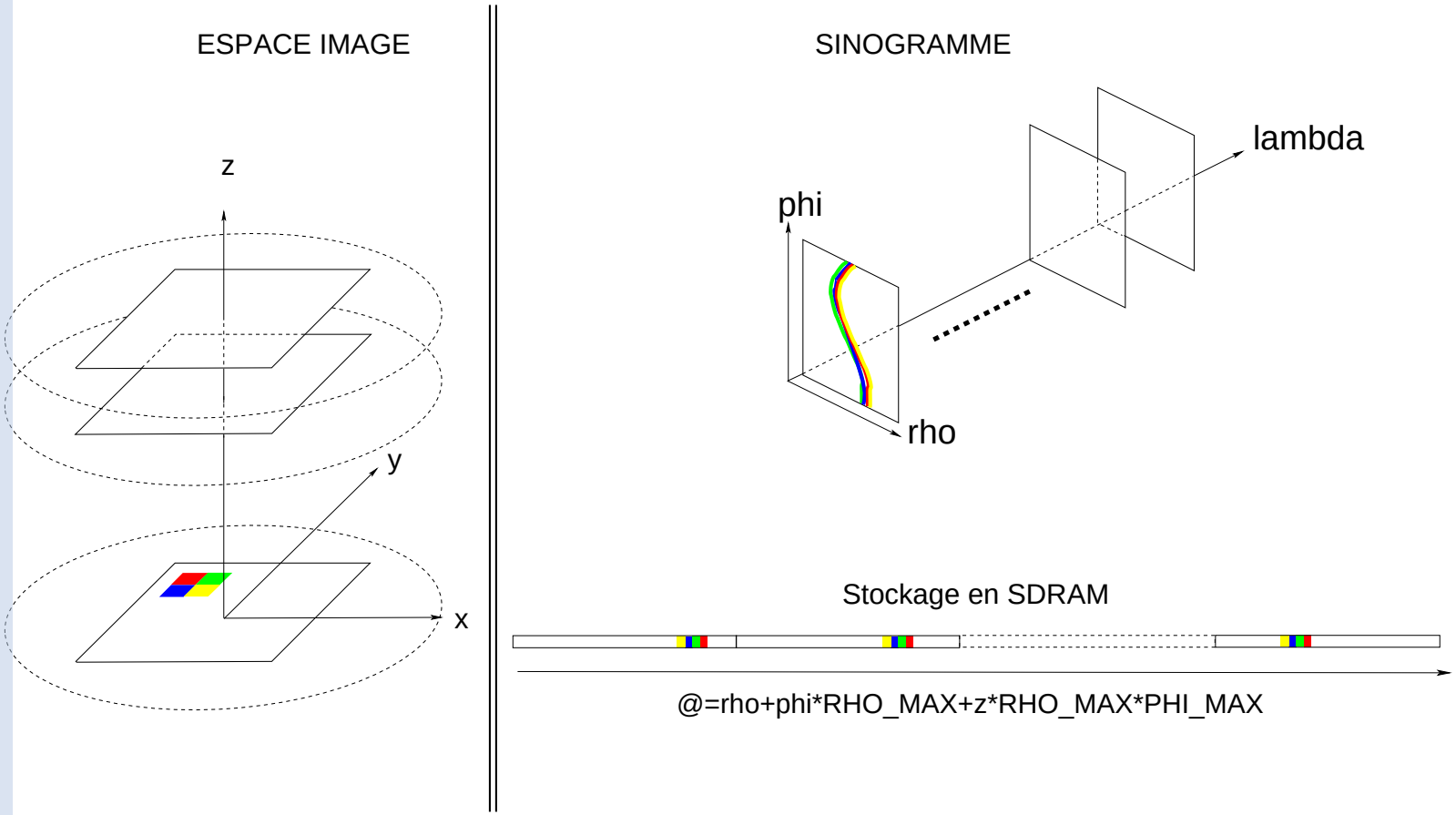
for Bloc in Volume **do**
  for $\mathtt{delta} = 0$ to $\mathtt{delta}_{\mathtt{max}} - 1$ **do**
    for $\mathtt{phi} = 0$ to $\mathtt{phi}_{\mathtt{max}} - 1$ **do**
      for $(\mathtt{xn}, \mathtt{yn}, \mathtt{zn})$ in Bloc **do**
        // *CALCUL DES COORDONNEES*
        $\mathtt{rho(phi)} = \mathtt{xn} \cdot \cos\phi + \mathtt{yn} \cdot \sin\phi$
        $\mathtt{lambda(phi, delta)} = ...$
        // *INTERPOLATION BILINEAIRE*
        $\mathrm{bin}_{interp} = C_{00} \cdot \mathrm{bin}_{00} + C_{01} \cdot \mathrm{bin}_{01}$ ...
        // *ACCUMULATION*
        $f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) = f^*(\mathtt{xn}, \mathtt{yn}, \mathtt{zn}) + \mathrm{bin}_{interp} \cdot J_\Delta$
      end for
    end for
  end for
end for

Estimation du taux de réutilisation des données

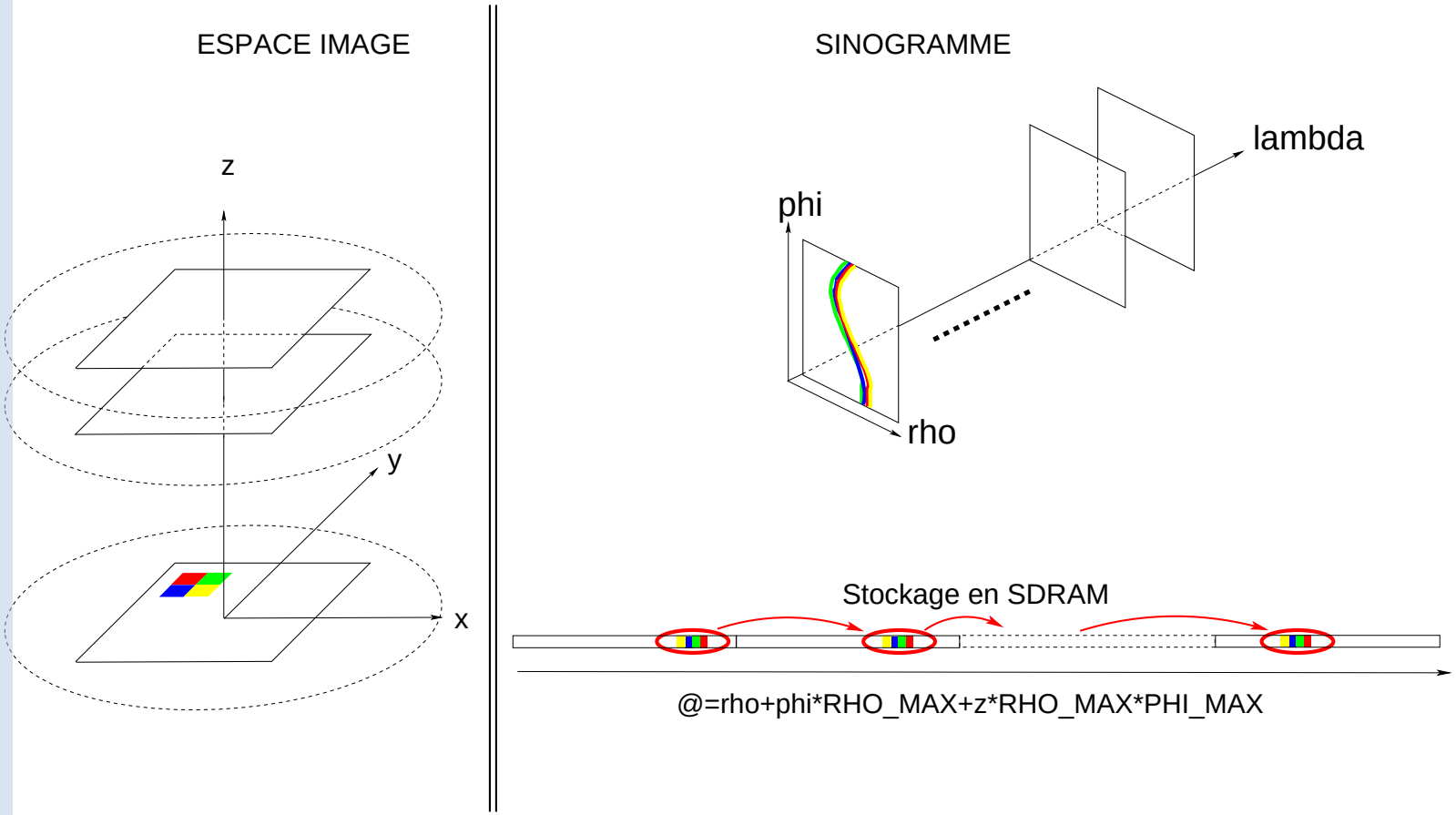


- Bloc de $16^2 \cdot 3$: $\simeq$ 9 sans interpolation bilinéaire ($\simeq$ 36 avec)

Mécanisme de préchargement mémoire



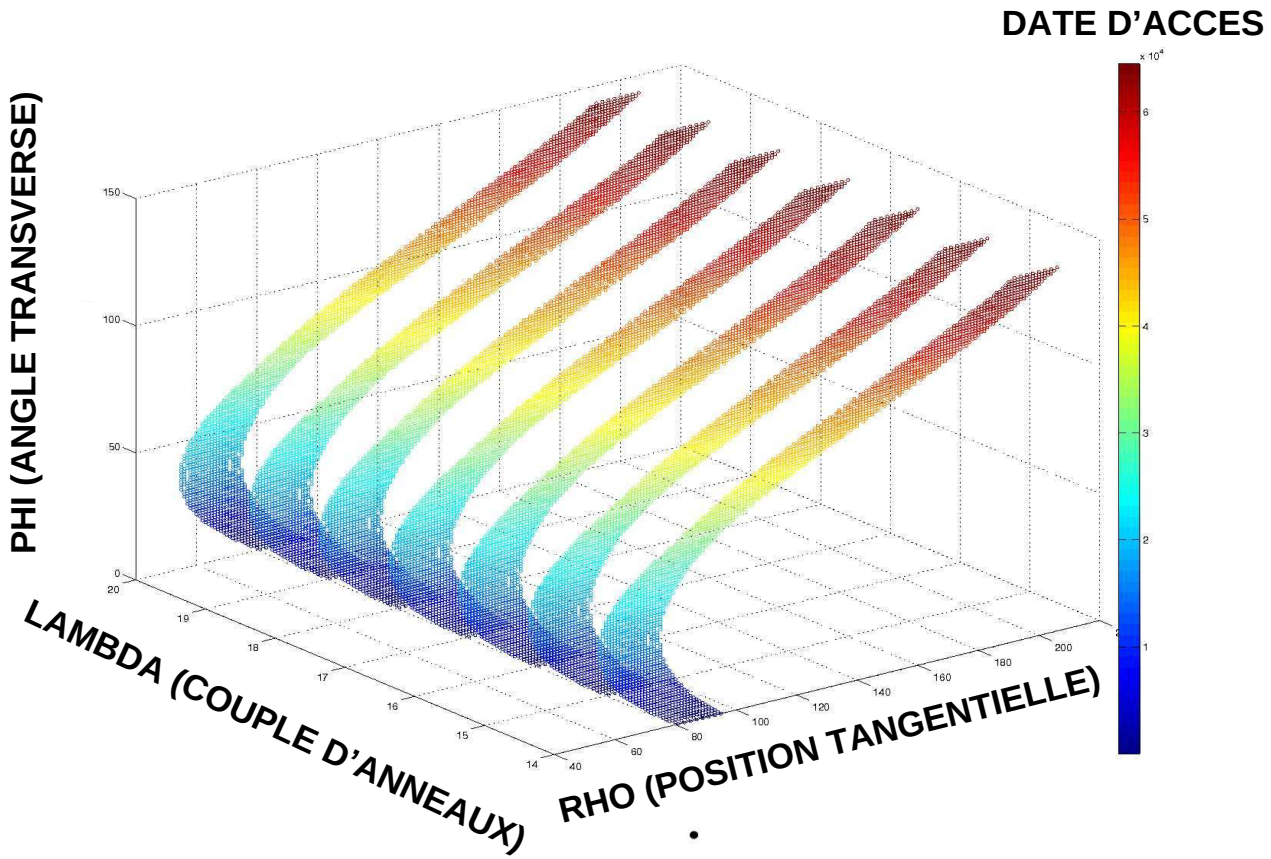
Mécanisme de préchargement mémoire



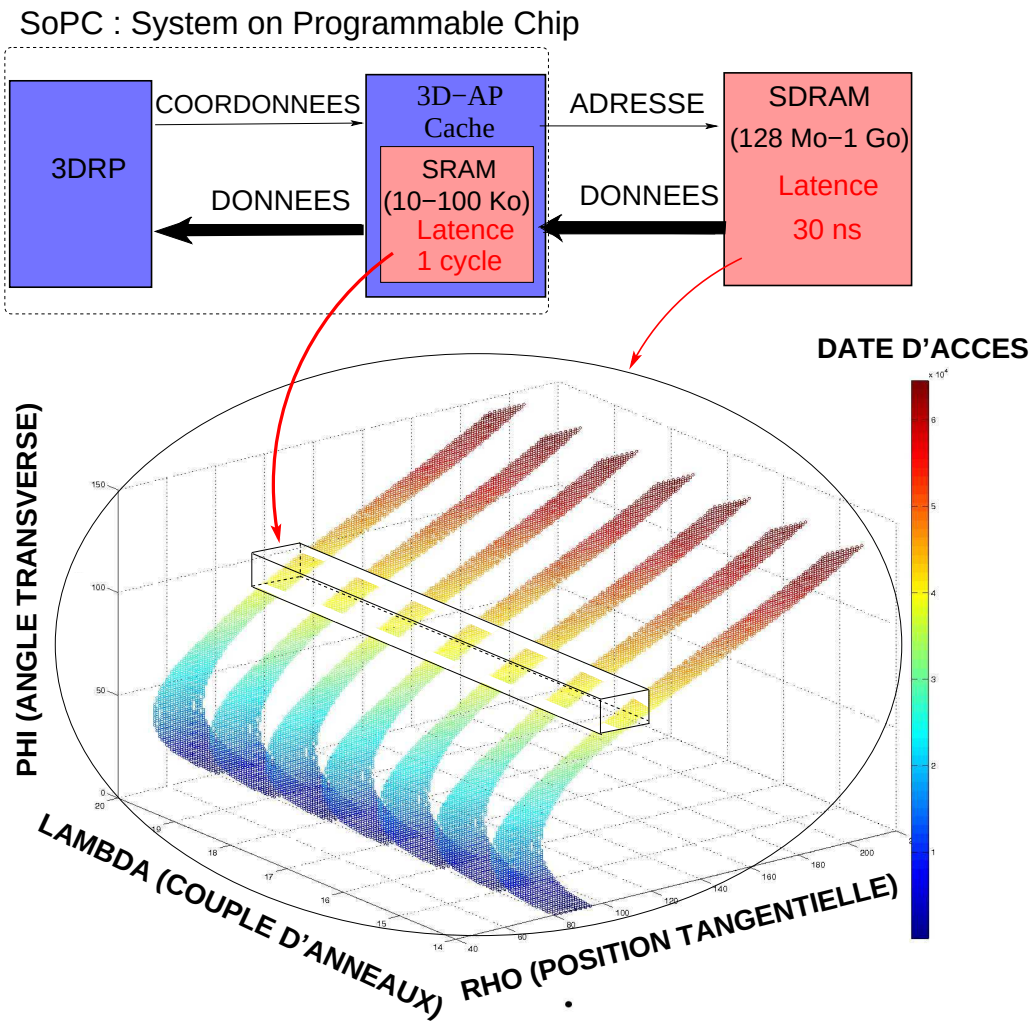
➲ Sauts dans l'espace mémoire non constant

Mécanisme de préchargement mémoire

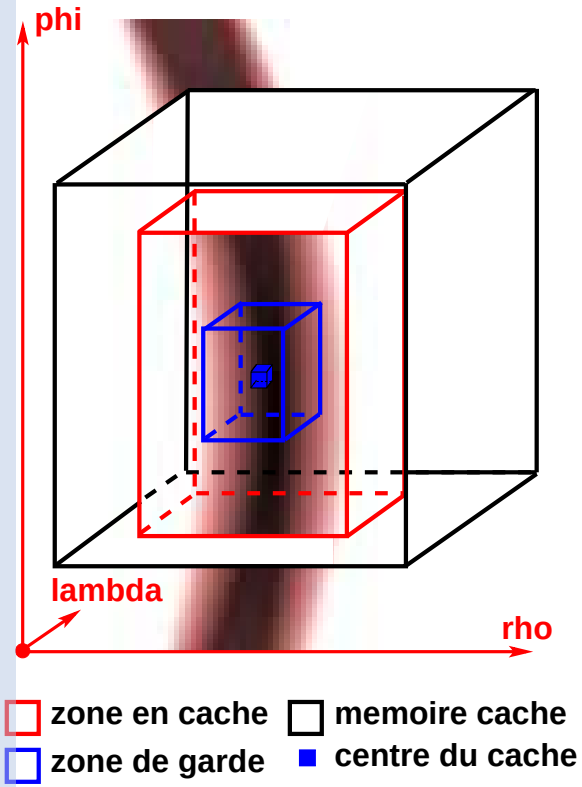
- *Parcours mémoire en sinusoïde 3D dans un segment*



Suivi de la sinusoïde 3D par un cache 3D-AP



Prédiction par analyse statistique



Coordonnées des accès mémoire précédents

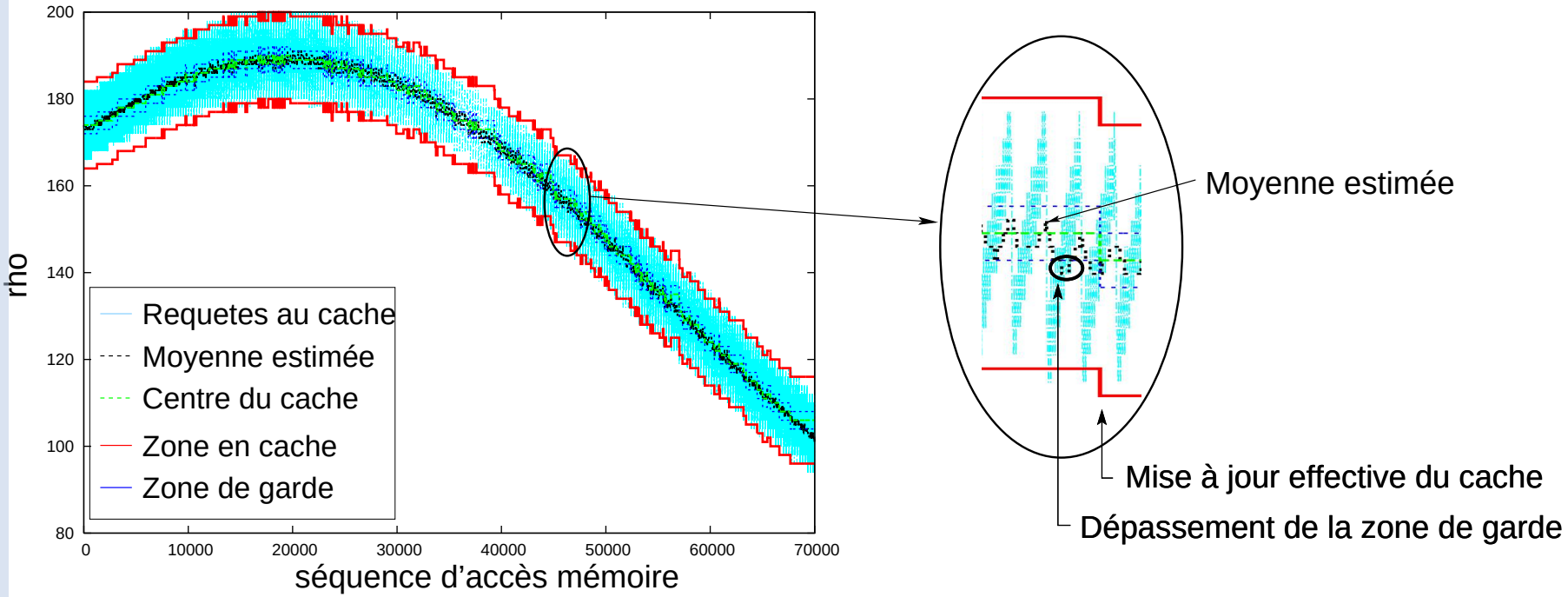
$$\overrightarrow{bin}(n) = \begin{pmatrix} phi(n) \\ rho(n) \\ lambda(n) \end{pmatrix}$$

Calcul de la moyenne des coordonnées

- Filtre IIR de premier ordre

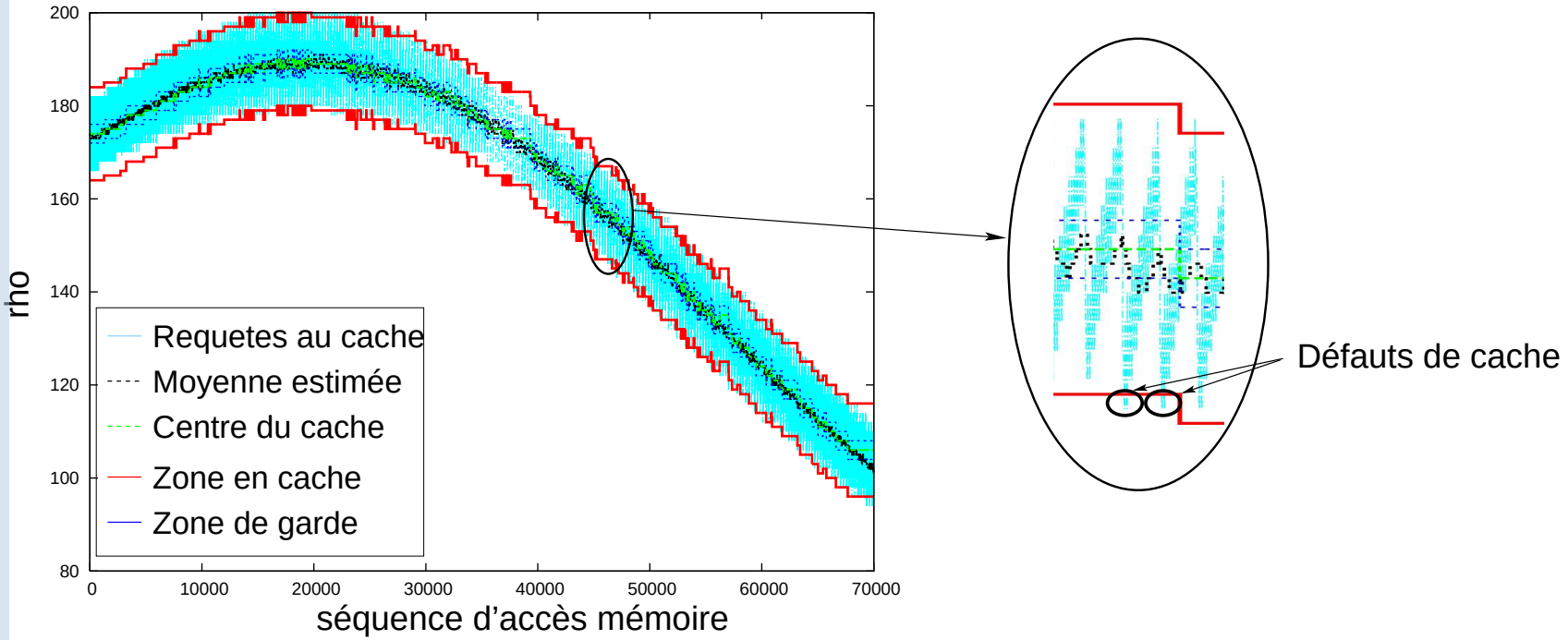
Suivi de la sinusoïde 3D

Mise à jour du cache



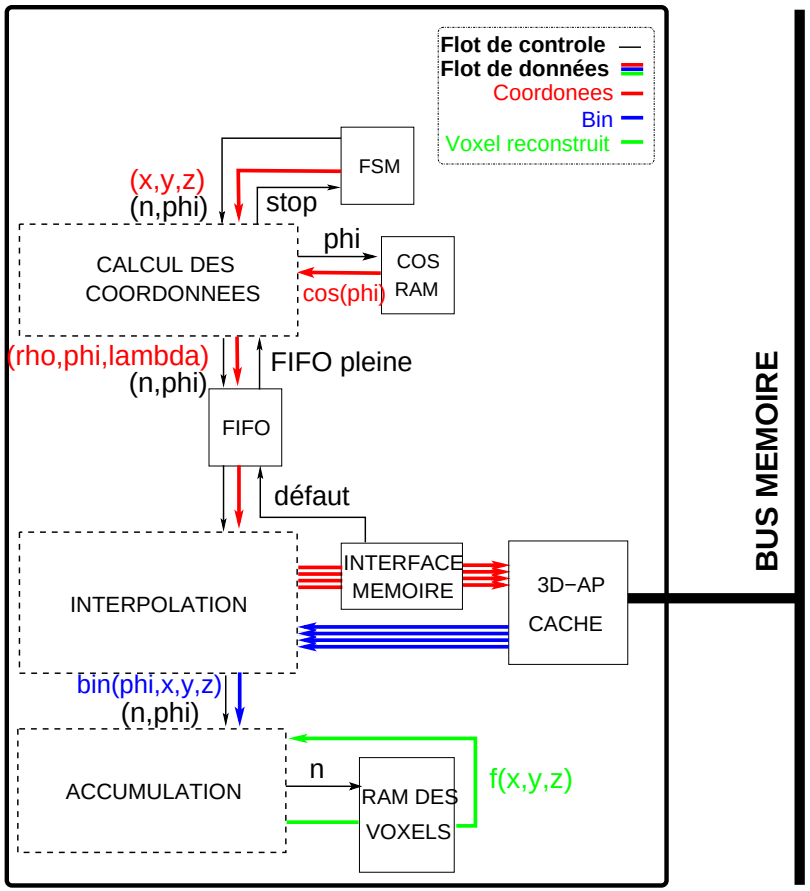
Suivi de la sinusoïde 3D

Traitement des défauts



Pipeline de rétroprojection 3D + Cache 3D-AP

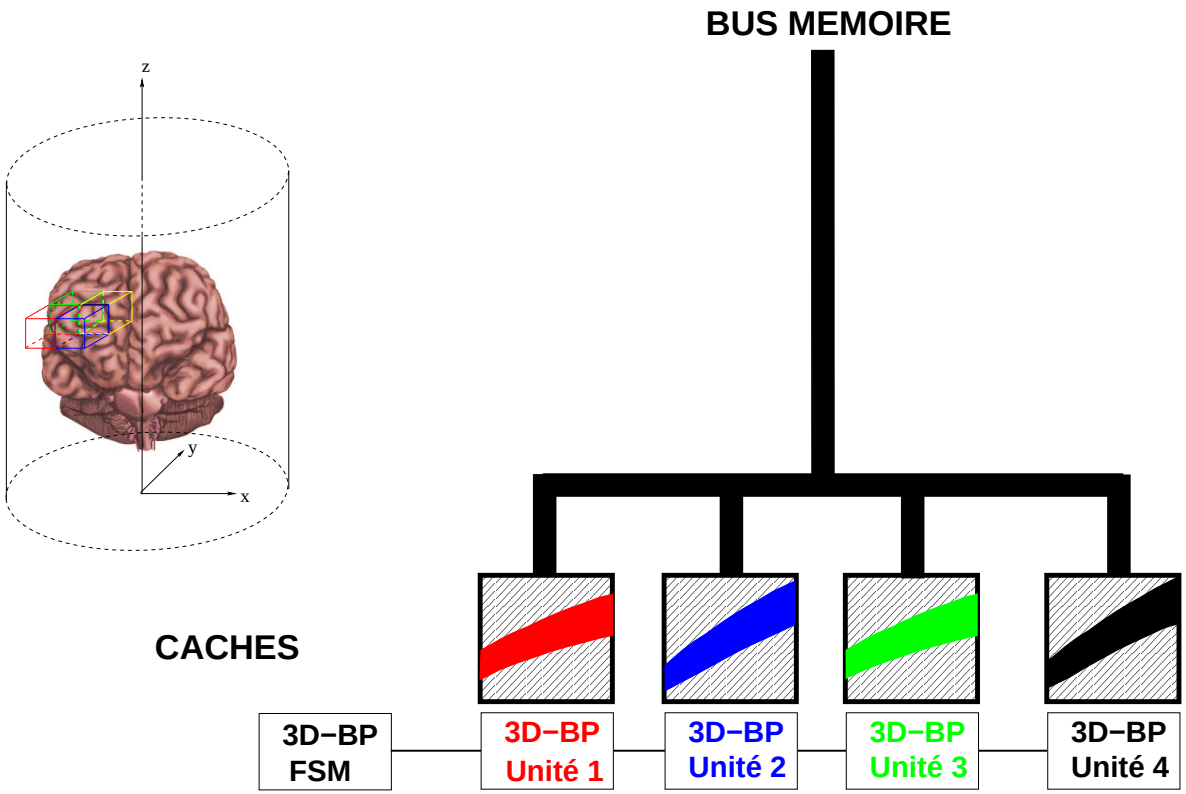
- **Objectif** : 1 cycle par mise à jour de voxel



Parallélisation

Données communes entre blocs adjacents

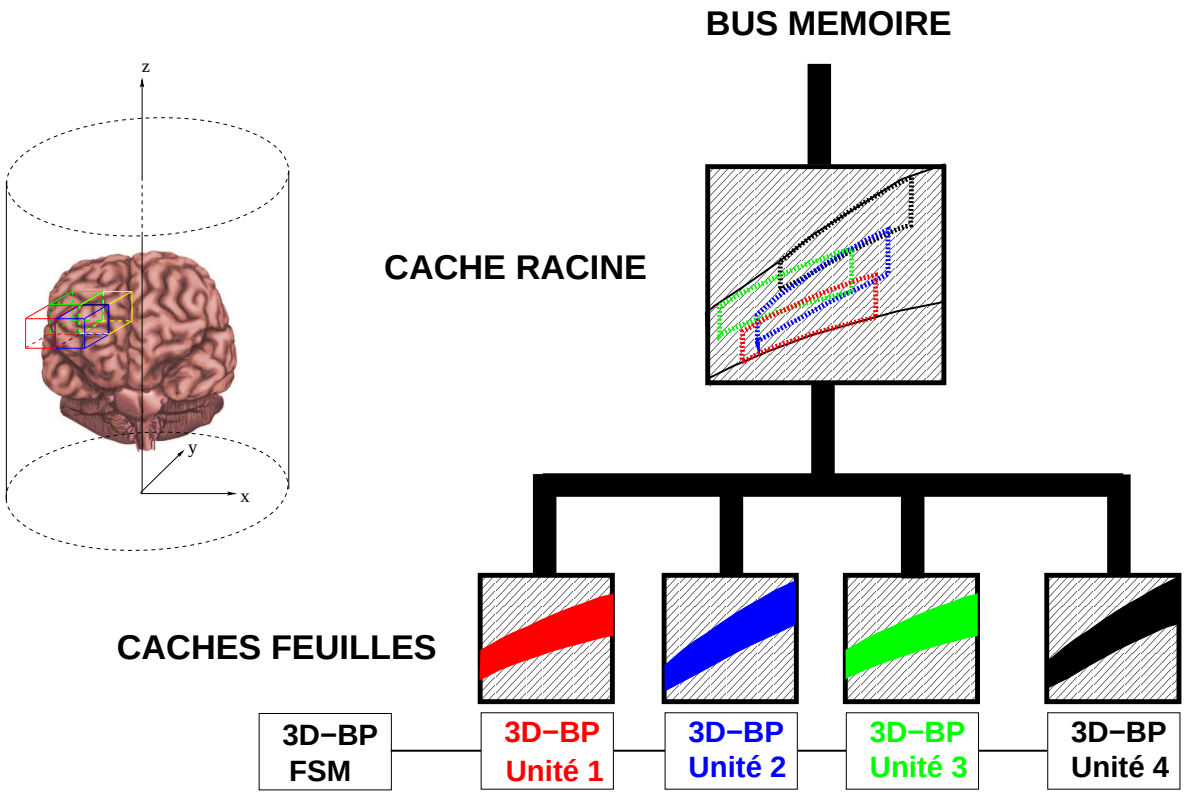
➲ *non réutilisées avec une architecture parallèle simple*



Parallélisation

Données communes entre blocs adjacents

➲ *réutilisées grace à une hiérarchie mémoire*



Protocole de mesure

Etude en simulation

- Premiers "calibrages" du cache
- Impossibilité de traiter des données de taille "réelle"

Prototypage de l'architecture 3P

- Implémentation sur un SoPC
- Simulateur du bus mémoire
  ➲ *Paramétrage de la latence et du débit mémoire*

Données utilisées

- Sinogrammes correspondant au scanner HR+ de Siemens
- Précorrections sur STIR (filtrage, correction en arc ...)

Mesures effectués sur la carte

❶ Comportement du cache 3D-AP

- η_{defaut} : taux de défaut de cache
- η_{cache} : taux de réutilisation des données mises en cache

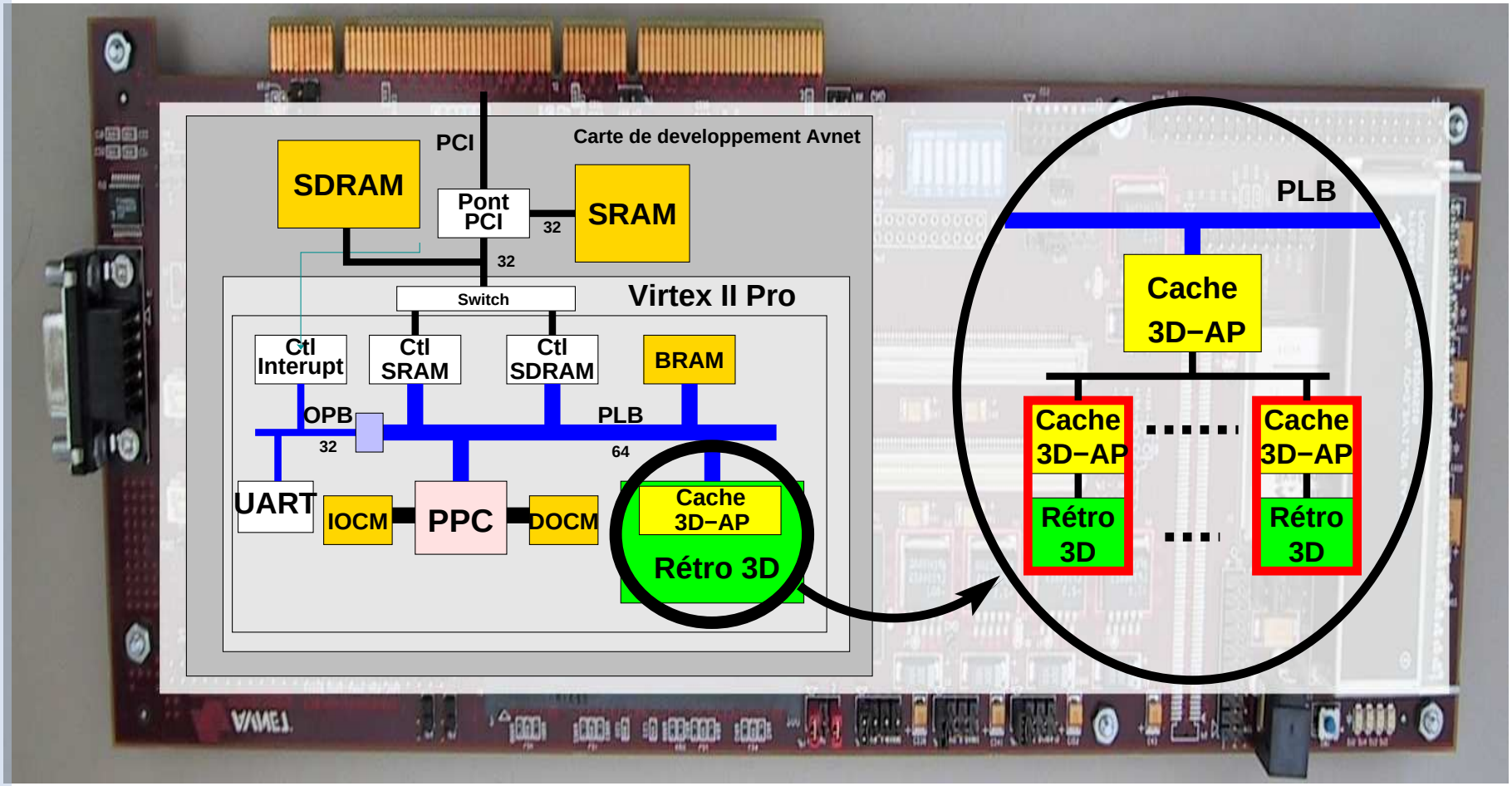
❷ Efficacité de reconstruction

- Nombre de cycles d'horloge par mise à jour de voxels

❸ Qualité de reconstruction

- EAM_r : Erreur Absolue Moyenne relative

Plateforme SoPC utilisée pour le prototypage



Ressources matérielles utilisées sur un Virtex 2 Pro

		1 unité	**4 unités**	**9 unités**
Rétro 3D	slices CLB	573	1 817	3 924
	Multiplieurs	12	48	108
Cache 3D-AP	slices CLB	672	2 830	4 804
	RAMs	2 Ko	24 Ko	36 Ko
Rétro 3D + Cache 3D-AP	slices CLB	1 245 (9.1%)	4 637 (32.9%)	8 728 (63.7%)

Xilinx 2VP30 (13 696 slices, 136 multiplieurs, 306 Ko RAMs)

Ressources matérielles utilisées sur un Virtex 2 Pro

		1 unité	**4 unités**	**9 unités**
Rétro 3D	slices CLB	573	1 817	3 924
	Multiplieurs	12	48	108
Cache 3D-AP	slices CLB	672	2 830	4 804
	RAMs	2 Ko	24 Ko	36 Ko
Rétro 3D + Cache 3D-AP	slices CLB	1 245 (9.1%)	4 637 (32.9%)	8 728 (63.7%)

Xilinx 2VP30 (13 696 slices, 136 multiplieurs, 306 Ko RAMs)

13 unités sur un Virtex 4 FX60

➲ limitation dûe au nombre de multiplieurs

Comportement du cache 3D-AP

Le cache suit correctement la sinusoïde 3D

- $\eta_{defaut} \simeq 0.1$ % pour un cache non hiérarchique
- $\eta_{defaut} \simeq 0.2$ % pour un cache hiérachique avec 8 unités

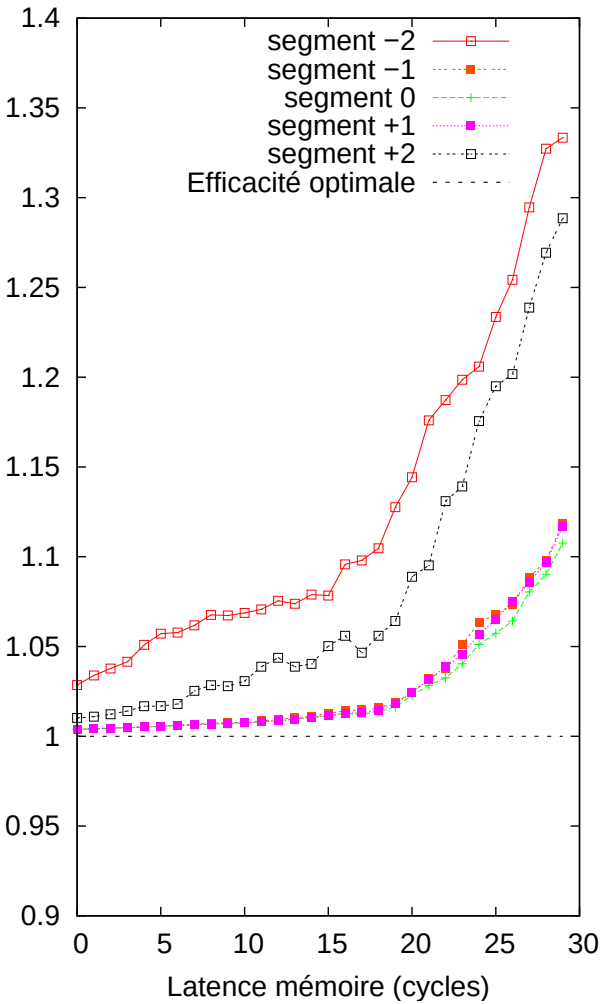
Difficulté à suivre "au plus près" une courbe 3D

- $\eta_{\text{cache feuille}} \simeq 10$ ($\simeq 36$ idéalement)

Hiérarchie mémoire utile à partir de 8 unités

- $\eta_{\text{cache racine}} \simeq 1.1$ pour 4 unités ($\simeq 3.5$ idéalement)
- $\eta_{\text{cache racine}} \simeq 1.7$ pour 8 unités ($\simeq 5$ idéalement)

Cycles/op en fonction des segments reconstruits



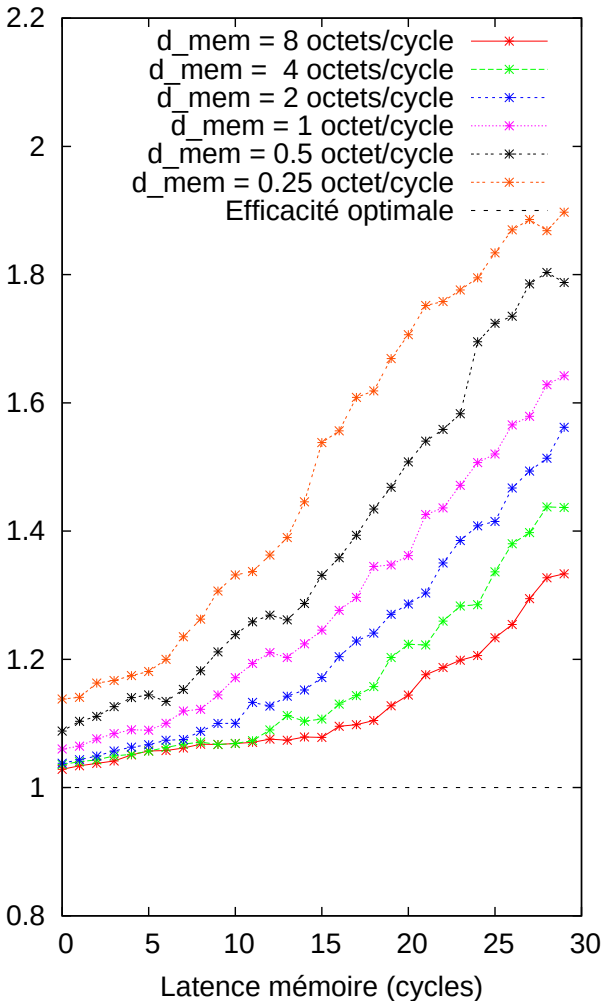
L'architecture 3P est efficace

- 1.05 cycles/op
  (latence de 30 ns @200 Mhz)

Légère sensibilité

- à la latence mémoire
- à la courbure 3D de la sinusoïde
  - ➲ Mise à jour du cache plus long
  - ➲ Les défauts bloquent le pipeline

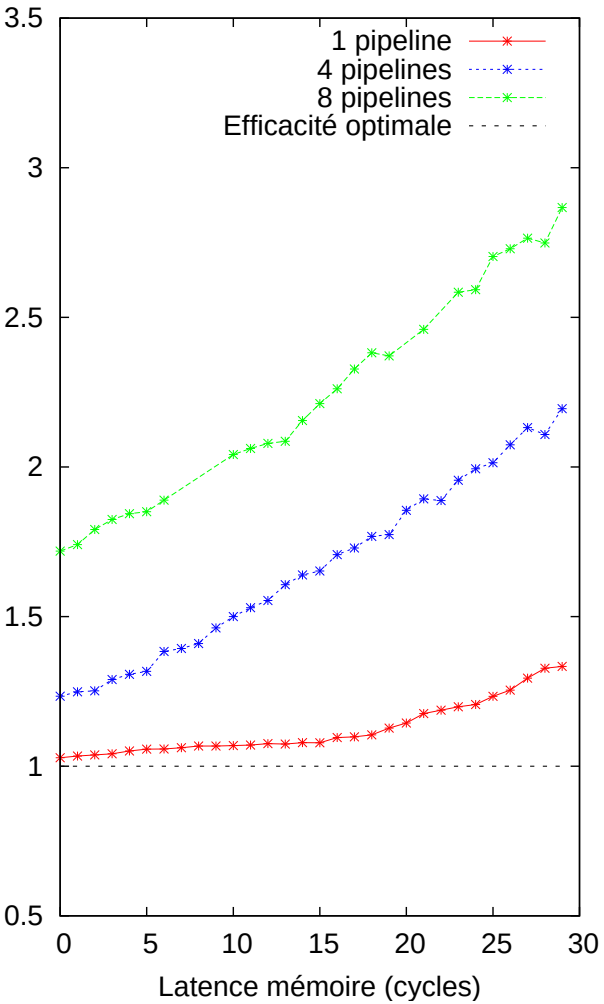
Cycles/op en fonction du débit mémoire



Robustesse face à la dégradation du débit mémoire

- Seulement 25 % moins efficace avec un débit 32 fois moins important

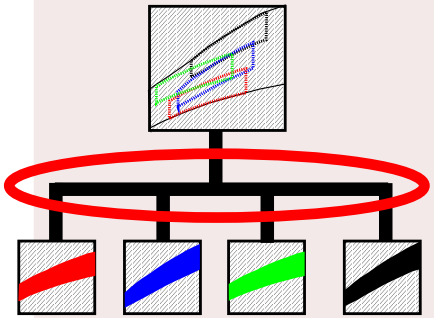
Cycles/op/unité pour 1, 4 et 8 unités



Facteurs d'accélération

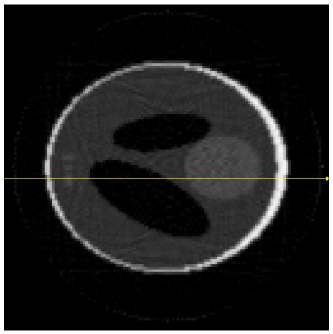
- 3.2 pour 4 unités
- 4.5 pour 8 unités

Conflits d'accès au cache racine

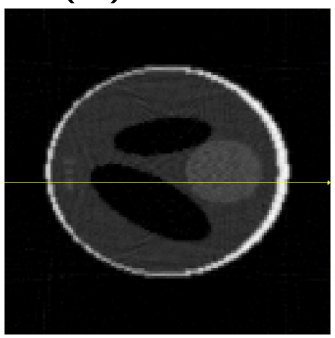


➲ Duplication de la mémoire racine
➲ Désynchronisation des calculs

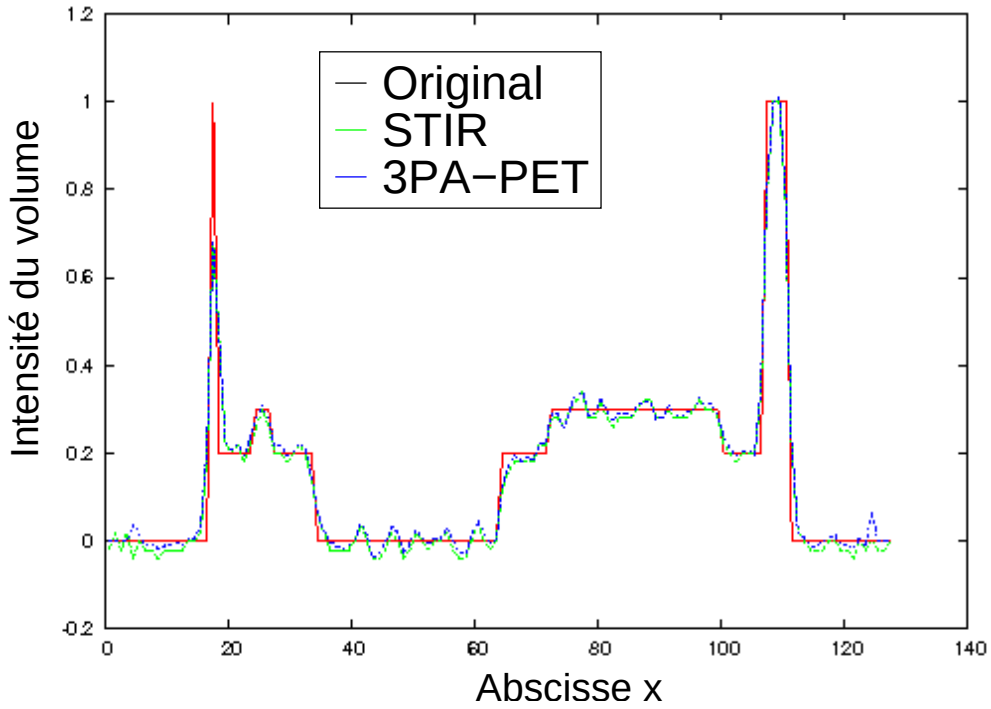
Volume Shepp Logan reconstruit en logiciel ("bit true")



(a) STIR



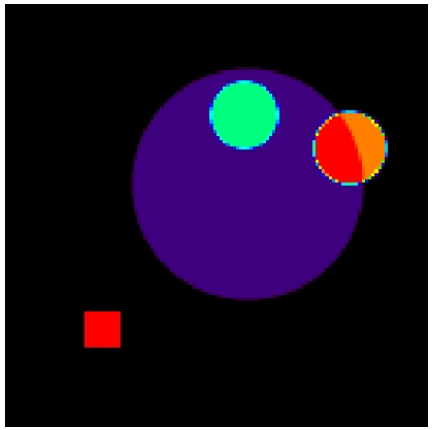
(b) 3PA-PET



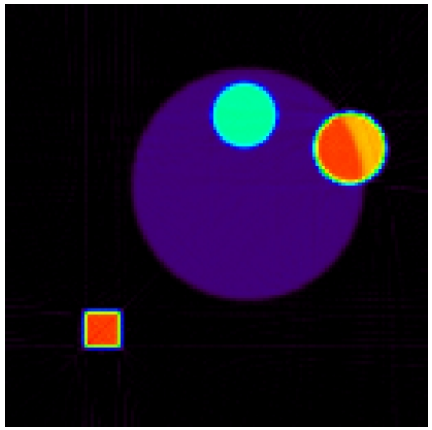
Comparaison des profils (a) et (b)

- Reconstruction de qualité satisfaisante ($EAM_r \simeq 1$ %)

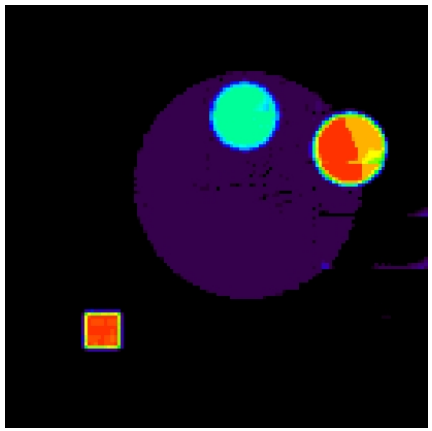
Volumes reconstruits sur la carte



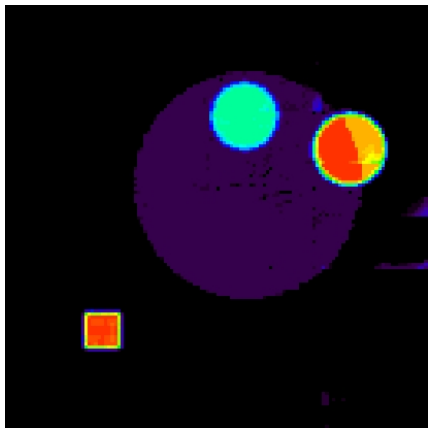
original



logiciel (virgule fixe)



carte (1 unité)



carte (8 unités)

Implémentation sur CPUs

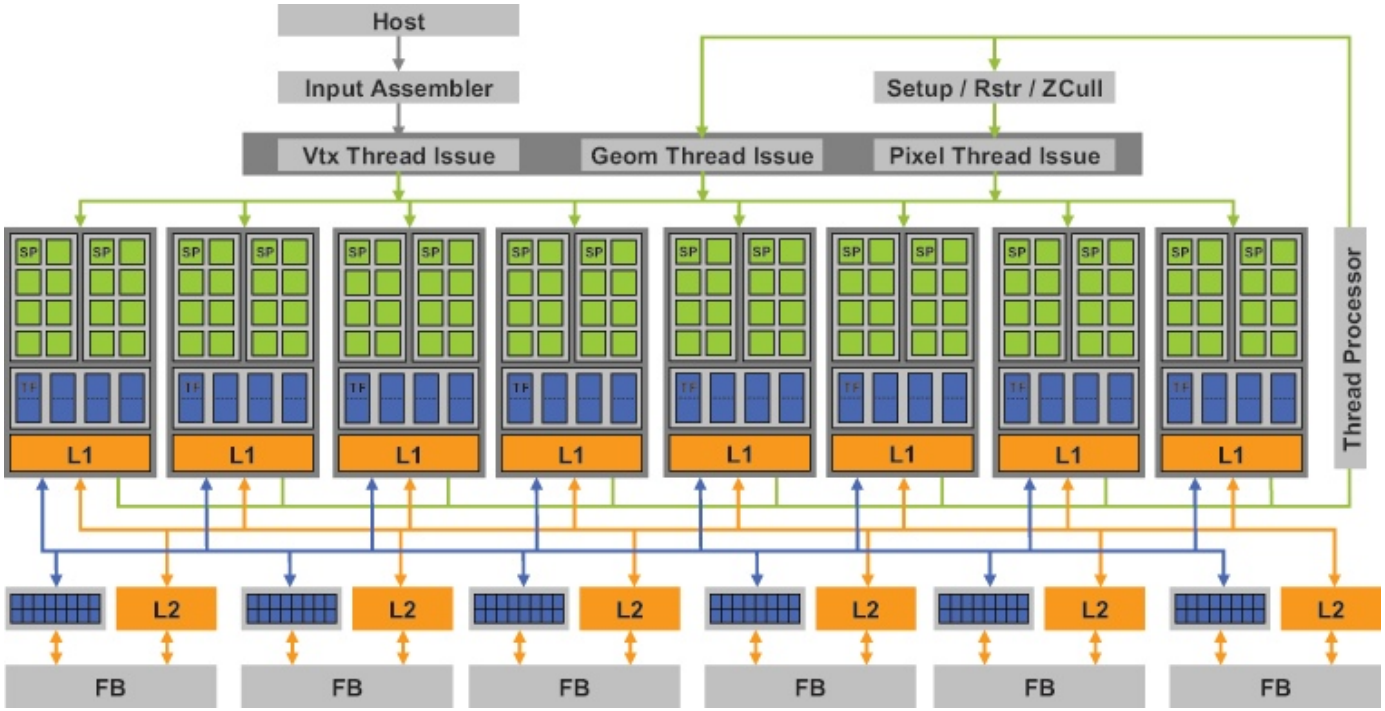
Caractéristiques du Pentium 4 et du bi-Xeon dual core

- Puissance de calcul : 3.2 GFlops (P4), 12 GFlops (bi-Xeon)
- Caches set associatifs (L1 : 16/32 Ko, L2 : 2 Mo)

Optimisations sur CPU

- Introduction localité spatiale et temporelle
  ➲ *accélération d'un facteur 3*
- Réduction du nombre d'opérations par mutualisation des calculs
  ➲ *accélération d'un facteur 7*
- Parallélisation avec la librairie *pthread* sur 4 coeurs (bi-Xeon)

Architecture des GPUs 8800 de Nvidia



Caractéristiques du GTS 8800

- Puissance de calcul : 260 GFlops (8*12 PEs)
- Caches 2D de texture (8 Ko)

Optimisations sur GPU

Parallélisation conservant la localité spatio-temporelle

- Reconstruction par blocs de voxels
- Reconstruction "en parallèle" des voxels adjacents dans chaque bloc de voxels

Réduction du nombre d'opérations par mutualisation des calculs

➲ *Accélération d'un facteur 2*

Temps de Reconstruction sur CPU/GPU/FPGA

Hardware		Nb Unités de Traitement	Temps	Cycles/Op /UT	Cycles/Op total
CPU	*Pentium 4 (3.2 Ghz, 6.4 Go/s)*	1	2.5 s	16	16
	bi-Xeon dual core (3 Ghz, 10.6 Go/s)	4	294 ms	7,12	1,78
GPU	GTS8800 (1.2 Ghz, 64 Go/s)	96	50 ms	12.9	0.135
FPGA	*Virtex 4 (200 Mhz, 0,8 Go/s)*	8	526 ms	1,7	0,21
ASIC	*5*3PA-PET (1.2 Ghz, 24 Go/s)*	40	27 ms	2.62	0,065

Temps de Reconstruction sur CPU/GPU/FPGA

Hardware		Nb Unités de Traitement	Temps	Cycles/Op /UT	Cycles/Op total
CPU	*Pentium 4 (3.2 Ghz, 6.4 Go/s)*	1	2.5 s	16	16
	bi-Xeon dual core (3 Ghz, 10.6 Go/s)	4	294 ms	7,12	1,78
GPU	GTS8800 (1.2 Ghz, 64 Go/s)	96	50 ms	12.9	0.135
FPGA	*Virtex 4 (200 Mhz, 0,8 Go/s)*	8	526 ms	1,7	0,21
ASIC	*5*3PA-PET (1.2 Ghz, 24 Go/s)*	40	27 ms	2.62	0,065

Algorithme 3D-RP et 3D-EM

Algorithme 3D-RP

- Algorithme analytique
- Etape de projection utilisée afin d'utiliser les segments inter-anneaux

Algorithme 3D-EM

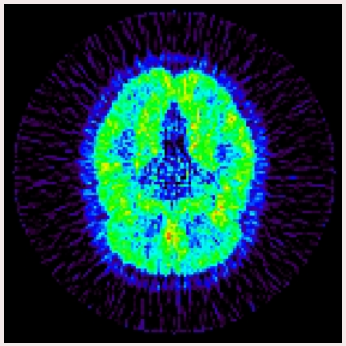
- Algorithme itératif bayesien
- Etape de projection fait partie intégrante du processus itératif

Paire matérielle de projection/rétroprojection

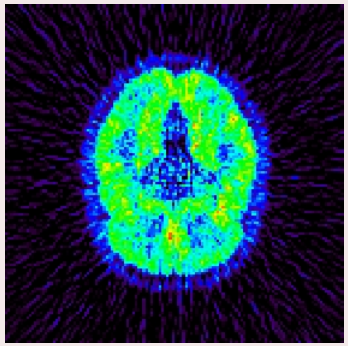
- Rétroprojecteur "voxel-driven" avec interpolation bi-linéaire
- Projecteur par lancer de rayon [Mancini07]

Qualité de reconstruction

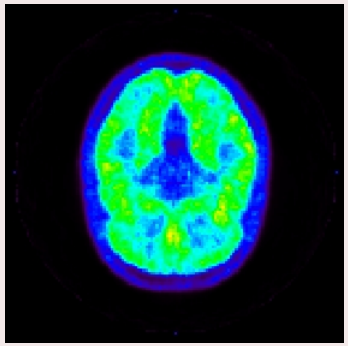
Reconstruction des volumes PET-SORTEO [Reilhac05]



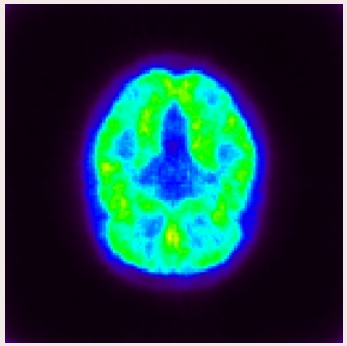
STIR (3D-RP)



ArchiTEP (3D-RP)



STIR (3D-EM)



ArchiTEP (3D-EM)

Quantification de l'écart de reconstruction avec STIR

- $\simeq 1\%$ pour l'algorithme 3D-RP
- $\simeq 3\%$ pour l'algorithme 3D-EM

Efficacité de reconstruction

Virtex 4 (200 Mhz) vs Pentium 4 (3.2 Ghz)

	Accélération par rapport à STIR			**Gain en efficacité par rapport à STIR**		
	Proj.	Rétro.	**Total**	Proj.	Rétro.	**Total**
3D-RP	6	17.5	**7.5**	95	300	**120**
3D-EM	3	12	**3.5**	50	200	**60**

Conclusion

Démarche d'Adéquation Algorithme Architecture

- Lever du verrou technologique constitué par le "mur mémoire"
  ➲ *Parcours mémoire en sinusoïde 3D + Cache 3D-AP*
- Généricité de la stratégie mémoire adoptée

Accélération de la rétroprojection 3D sur CPU/GPU/FPGA

- Fort impact fort de la localité spatio-temporelle
- GPU plus rapide mais architecture 3P plus efficace

Vers une reconstruction de meilleure qualité

➲ *algorithmes 3D-RP et 3D-EM*

Perspectives

Amélioration de la parallélisation des calculs

- Parallélisation sur une puce SoC
- Parallélisation sur carte multi SoPCs
  ➲ *carte du projet ArchiTEP : 1+6 Virtex 4*

Accélérations Algorithmiques

- OSEM, sous échantillonage, méthodes "divide and conquer"

Tomographie CT

- Rétroprojection à faisceaux coniques

Merci de votre attention !