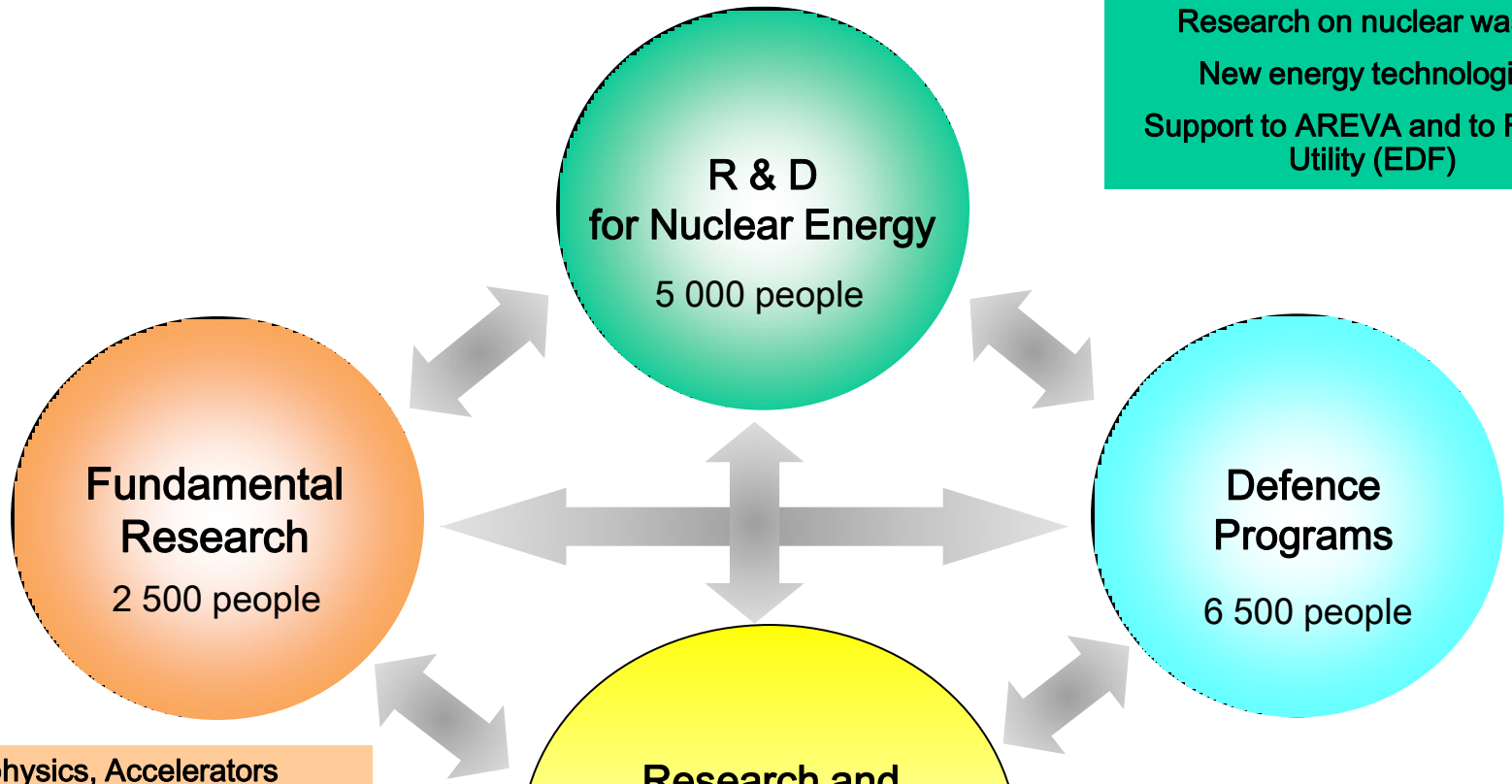
Contrôle Non Destructif : Implantation d'algorithmes sur GPU et multi-coeurs

Gilles Rougeron
CEA/LIST
Département Imagerie Simulation et Contrôle

CEA



Nuclear systems for the future

Research on nuclear wastes

New energy technologies

Support to AREVA and to French Utility (EDF)

Nuclear physics, Accelerators

Thermonuclear fusion (ITER)

Astrophysics

Climate and environmental science

Chemistry and radiation-matter interactions

Applying nuclear-based technologies to health

Micro-technology and nanoscience

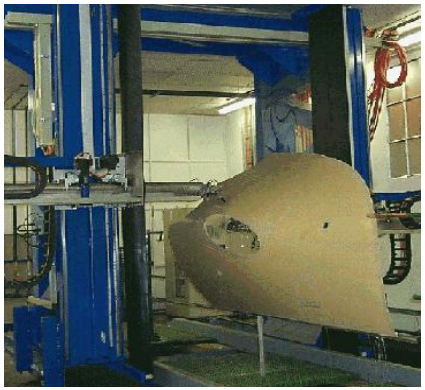
Embedded systems, advanced softwares, sensors and signal processing

Health technologies : functional imagery, bio molecule

LETI, LITEN, LIST

Le Contrôle Non Destructif

Techniques non invasives pour la détection de défauts critiques dans des pièces ou structures industrielles



- Contrôles en fabrication ou contrôles en maintenance
- Concernent tous secteurs industriels à des degrés divers: Energie, aéronautique, pétrochimie, transports, sidérurgie…
- De forts enjeux économiques + enjeux sécurité du public
- Des techniques variées dont les principales: Ultrasons, Méthodes électromagnétiques (courants de Foucault, flux de fuite, etc…), Radiographie X ou γ
- Secteur dynamique en pleine évolution, forte demande de R&D

La Simulation des CND

Principaux enjeux

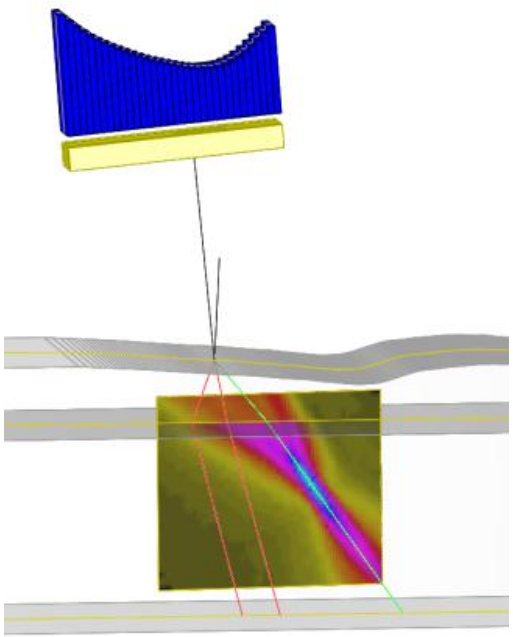
- Conception des méthodes, design des capteurs
- Evaluation et qualification des méthodes
- « Contrôle virtuel » dès le bureau d'études
- Aide au diagnostic, inversion
- Formation

Développement de CIVA

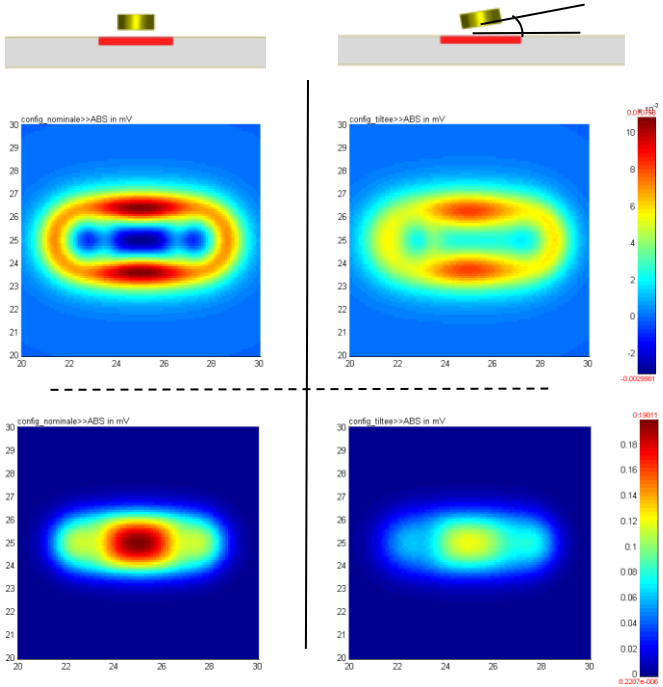
- IHM métiers, outils CAO, visualisation 3D
- Analyse de données simulées ou issues d'acquisition
- Plate-forme d'intégration (partenariats académiques, plugin)
- Multi-techniques CND : *UltraSons, Electromagnétique, rayons X.*
- Codes en Java / C++ / Matlab – OS Windows

CIVA

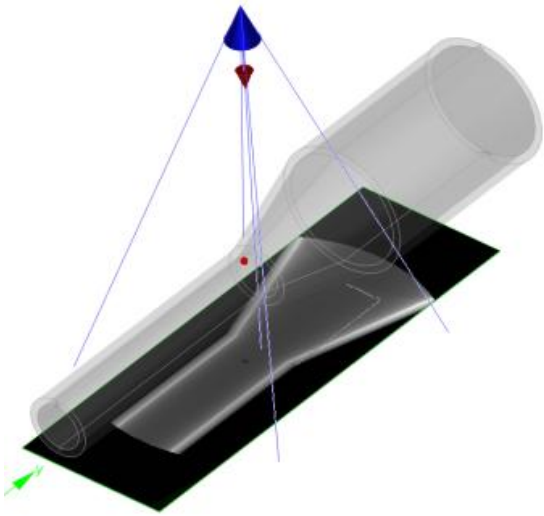
Ultrasons



Electromagnétisme



Rayons X



Développement de CIVA

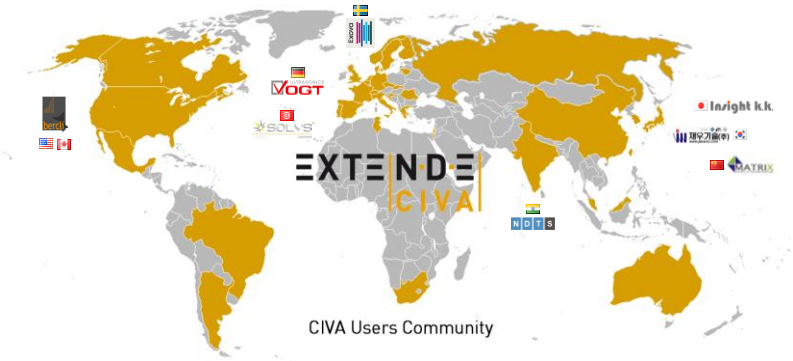
Sorties régulières de versions intégrant la R&D récente

25 développeurs permanents: Modélisation, algorithmie, informatique

1 structure de tests et validation, et un réseau de distribution

CIVA V10

- **144** clients répartis dans **34** pays.
  - Industriels (Nucléaire, Transport, Pétrochimie, Métallurgie)
  - Laboratoires, Universités
- Plus de **200** licences actives
- **100** personnes formées à CIVA en 2010



Intégration dans un logiciel industriel

Contraintes liées à la diffusion de Civa

Utilisateurs

Validité des méthodes
Simplicité d'utilisation
Matériel standard (PC Windows + GPU toute marque)

Développements

Code doit être : pérenne (projet démarré il y a 15 ans, version 11 en développement), maintenable, robuste.
Documentation – 1.5Millions de lignes de code
Partage de la connaissance en équipe

Intégration dans un logiciel industriel

Demande forte des industriels pour des calculs rapides

Démocratisation des moyens matériels pour le HPC

Architectures hardware

CPU multi core/ many core
GPU
Clusters

Outils logiciels (adaptation-modification du code existant)

Cuda, OpenCL

OpenMP, Intel tbb, Microsoft Parallel Pattern Library

Calcul distribué (ProActive)

Générateurs de code
HMPP

Calcul Parallèle dans CIVA : 3 niveaux de parallélisme

Travaux en cours

Clusters

Calculs de lots de fichiers distribués sur un serveur de calcul Multi-techniques

PC Multi-coeurs

Parallélisation des codes de calcul : US/EM/RX

GPU

Traitements et Reconstruction (US/RX)

Projets en cours : ITOC et OPARUS

Interface Tomographique pour le CND

- Laboratoire Recherche Informatique – Université de PARIS-SUD 11
- CMAP – Centre Mathématiques Appliquées de l'école Polytechnique
- CEA LIST

Intégration dans la plateforme Civa de code de calcul en tomographie X avec outils d'analyse. Reconstruction sur GPU.

Optimisation et Parallèlisation pour l'Analyse et la Reconstruction du CND par UltraSons

- EADS/EDF
- M2M/CAPS Entreprise
- CEA/LIST
- Institut d'Electronique Fondamentale – Université de PARIS-SUD 11

Reconstruction rapide dans le cadre de contrôles industriels par utilisation des GPU.

True Cumulated View

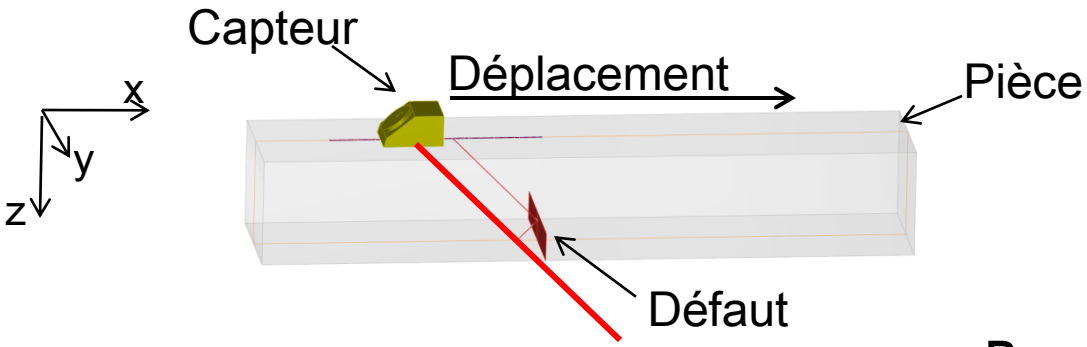
Analyse de données après simulation ou acquisition **ultrasonore**

Reconstruction = repositionnement des échos (les défauts) dans l'espace réel (la pièce) dans un but de caractérisation.

Besoin de performance traiter de gros volumes de données, interactivité contrôle sur site.

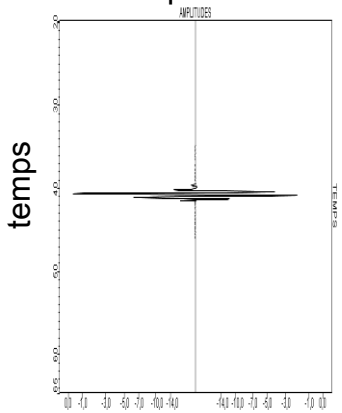
Thèse A. Pédron/Dir L. Lacassagne
(TCV présenté PARCO 2011)

Principe de la reconstruction « trajet »

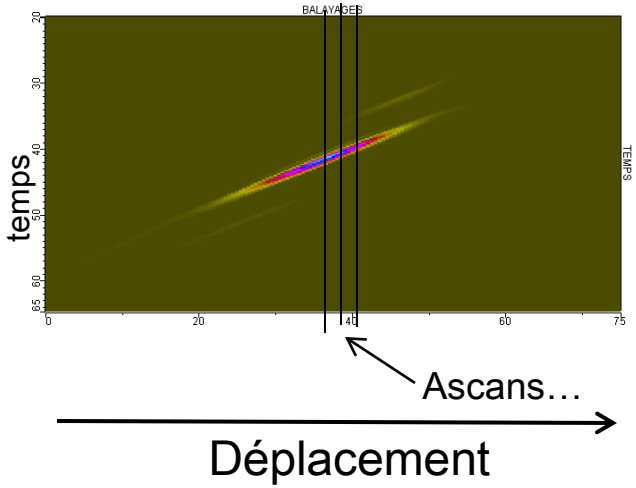


Ascan

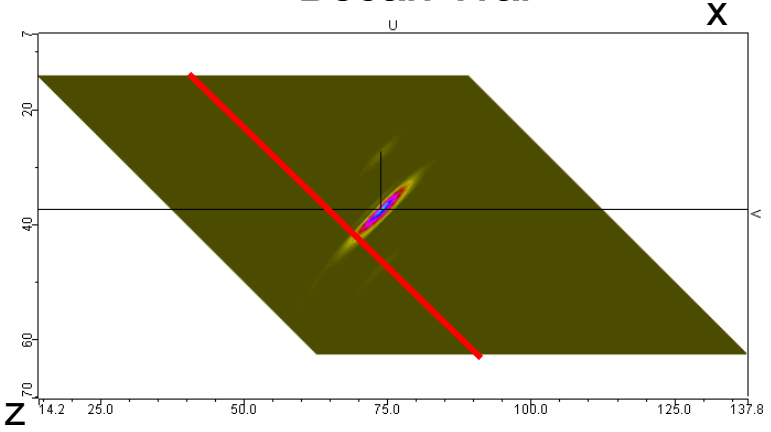
Amplitude



Bscan

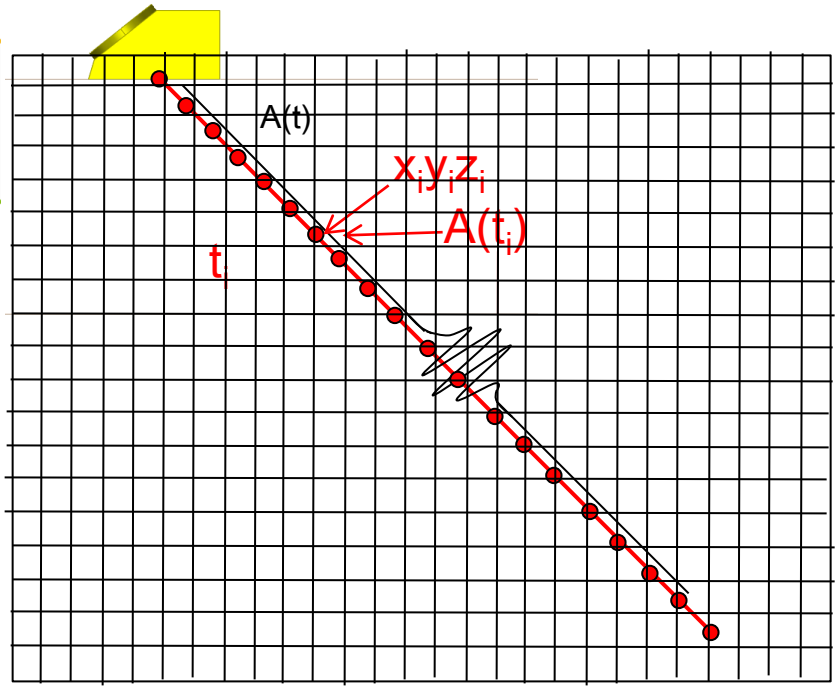


Bscan Vrai

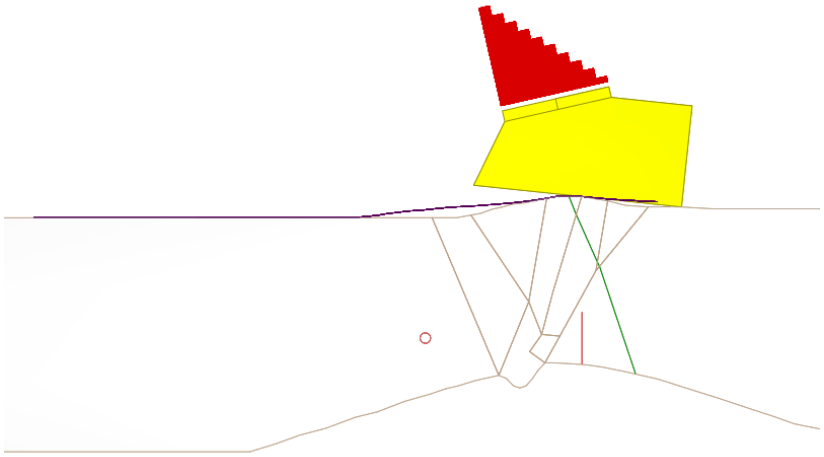


Direction de propagation approximée par un rayon (axe focal) = trajet
Vitesse de propagation connue du matériau.

Principe de la reconstruction « trajet »



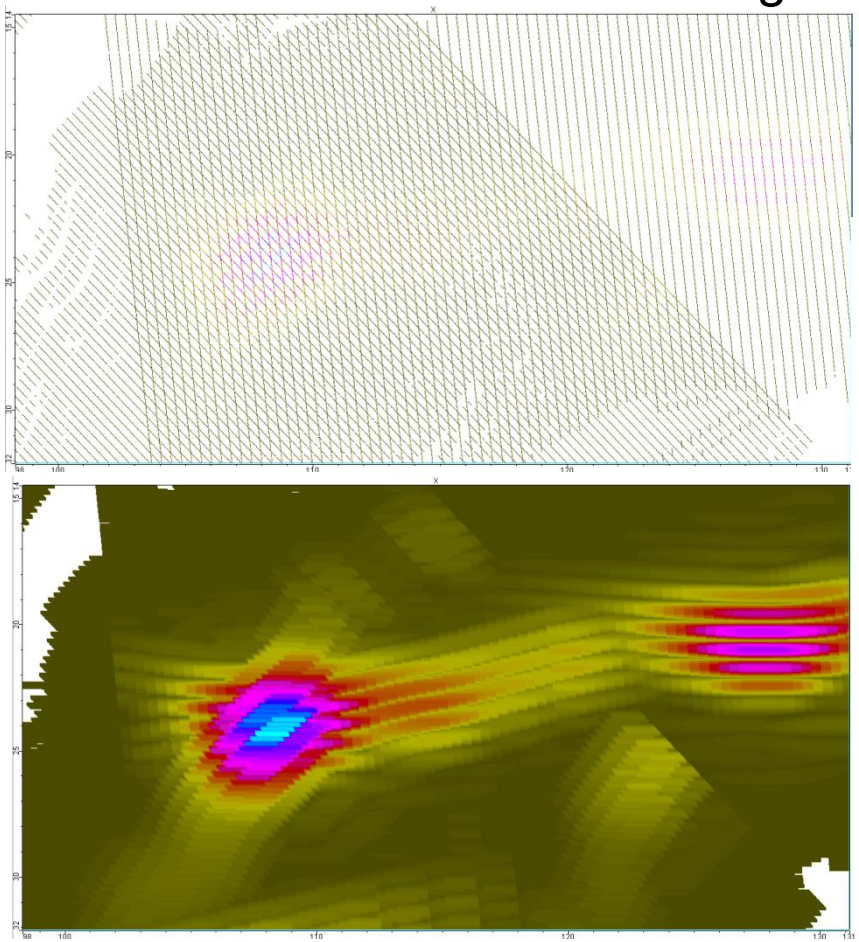
Donnée Trajets : non régulière !
Trajet = suite de segments de droite
D'un tir à l'autre pas de cohérence spatiale



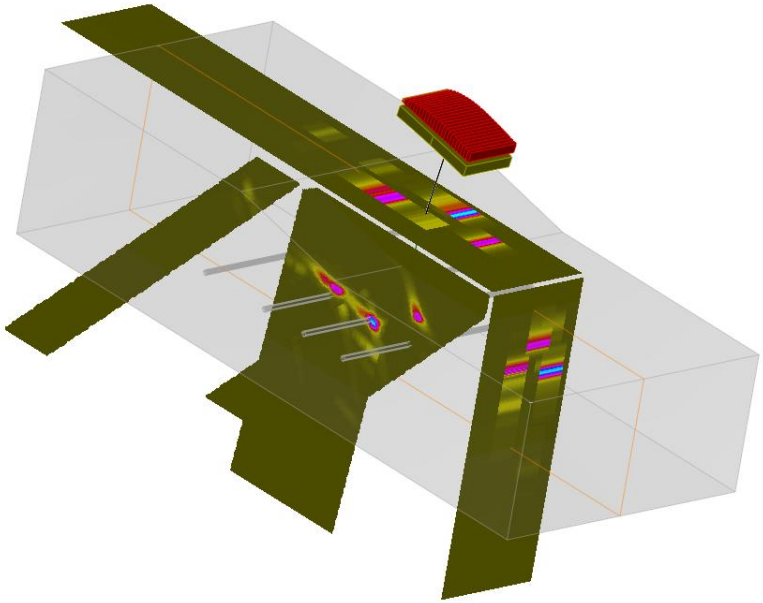
Projection sur un plan
Domaine de reconstruction discrétisé en une image pixelisée
$A_{pixel} = \max(A_{pixel}, A(t_i))$ (accumulation)

True Cumulated View : principe

Dilatation = max sur un voisinage



Ensemble des trajets =
Volume de données 3D



Taille voisinage dépend :
- Paramètres liés au contrôle
- Définition de l'image

TCV sur 3 plans

True Cumulated View : algorithme

Données Entrée

- Signaux temporels A(t)
- Trajets (x,y,z ,t)
- Aire reconstruction : grille 2D

Donnée Sortie

- Image en fausse couleur

Algorithme 1

Pour tous les trajets

Pour chaque pas de temps du signal associé au trajet t_i

Calcul position (x_i,y_i,z_i) par **interpolation** sur le trajet

Projection du point sur le domaine de reconstruction – max(pixel,$A(t_i)$)

Dilatation
Mise à jour des valeurs amplitudes des pixels de la grille = max($A(t_i)$, A_{pixel}) sur un voisinage

Visualisation image amplitude

True Cumulated View : algorithme

Optimisation

La diffusion de la valeur du max amplitude sur un voisinage peut s'effectuer en post traitement.

Dilatation en post-traitement global sur l'image

Algorithme 2

Pour tous les trajets

Pour chaque pas de temps du signal associé au trajet t_i

Calcul position (x_i,y_i,z_i) par **interpolation** sur le trajet

Projection du point sur le domaine de reconstruction – max(pixel,$A(t_i)$)

Dilatation = Attribution pour chaque pixel de l'image du max amplitude déterminé sur un voisinage.

Visualisation image amplitude

True Cumulated View : Implémentation CPU

Différentes implémentations et Benchmarks

OpenMP CPU
CUDA / NVIDIA GPU Tesla

OpenCL (CPU et GPU)

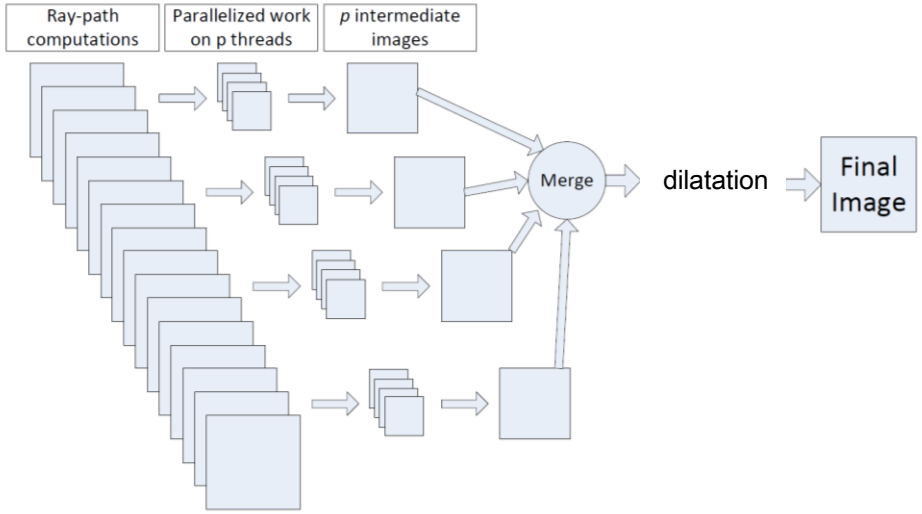
True Cumulated View : Implémentation CPU

C++ et OpenMP (Visual C++ 2010)
Chaque thread travaille sur un trajet à la fois et crée une image intermédiaire d'amplitudes max. Puis fusion images.

Pas de concurrence !

Etapes Interpolation-Projection **parallélisée**
Fusion image **parallellisée et vectorisée**
Dilatation **parallélisée**

Algorithmes 1 et 2(optimisé) codés.



True Cumulated View : Implémentation GPU

Développements Cuda Toolkit 3.2 - JCuda

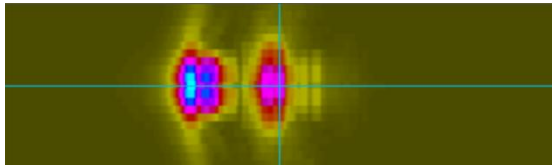
1. **Parallélisation sur les trajets répartis sur thread GPU.**
   PBM : concurrence lors du calcul de max sur image.
   => utilisation des atomics

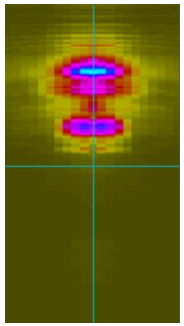
2. **Parallélisation sur les pixels de l'image**.
   Par pixel, recherche des trajets contributeurs.
   => tri des trajets

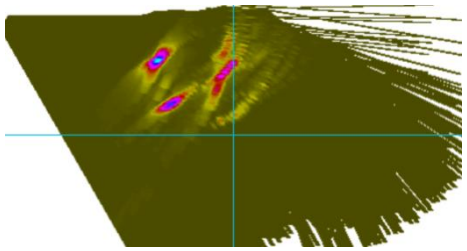
Choix approche 1 (atomicMax Cuda)

Deux versions :
- Algorithme 1 : Dilatation à la volée
- Algorithme 2 : Dilatation en post-traitement

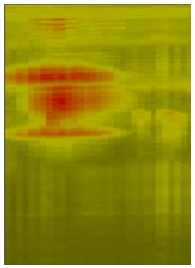
True Cumulated View : Résultats test







EXZ (15000 trajets, signaux 2000 échantillons = 30M points)



PMF (150000 trajets, signaux 1000 échantillons = 150M points)

	CPU1		CPU2 algo optimisé		GPU1		GPU2 algo optimisé		GPU2 optim/ CPU1	GPU2 optim / CPU2 optim
	1 thread	4 threads	1 thread	4 threads	C1060	C2070	C1060	C2070	slowest / fastest	fastest / fastest
View type	EXZ dataset / 800 × 800 image reconstruction									
Side View	5.02	1.32	1.16	0.32	0.92	0.34	0.14	0.072	×**69**	×**4**
Top View	43.12	11.37	1.19	0.37	10.36	1.71	0.21	0.068	×**631**	×**5**
Front View	14.07	3.67	1.22	0.36	6.19	1.86	0.24	0.077	×**180**	×**5**
	PMF dataset / 800 × 800 image reconstruction									
Side View	18.6	4.9	5.9	1.6	2.4	1.3	0.7	0.33	×**69**	×**5**
Top View	60.3	15.9	5.9	1.5	10.8	2.8	0.8	0.15	×**388**	×**10**
Front View	62.2	16.1	6.0	1.6	20.0	11.1	1.0	0.27	×**197**	×**6**

Temps exécution en s

True Cumulated View : Analyse Algorithme optimisé

GPU

Gain avec nouvelle génération (C2070/C1060) x2 à x5

Irrégularité par rapport aux données et paramètres :

- Variation de la concurrence (atomics)
  - Localisation de la projection des trajets
  - Taille image
- Accès mémoire aléatoire

Exécution kernel étalement <10ms même pour filtres de grande taille

CPU OpenMP

Régulier par rapport aux données et aux paramètres
Bon passage à l'échelle
Opération étalement peut être bien plus lente que sur GPU pour voisinages de grand taille.

True Cumulated View : Portage vers OpenCL

Portage Cuda vers OpenCL de l'algorithme optimisé
(stagiaire M2 : Victor Barbillon)

JOCL – Nvidia OpenCL (Cuda Toolkit 3.2) / AMD Accelerated Parallel Processing (APP) SDK 2.4

Tests de perfomance

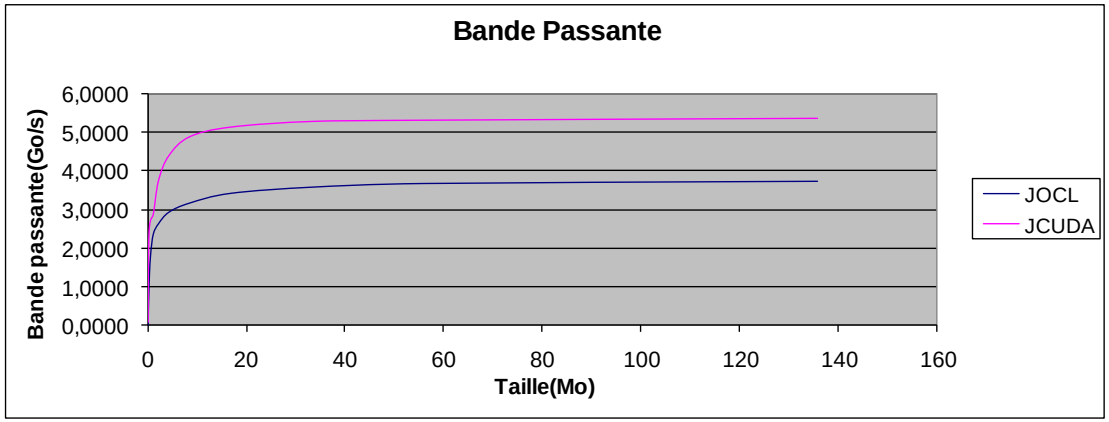
1. Cuda vs openCL NVIDIA C 2050
2. OpenCL GPU (NVIDIA C2050) vs OpenCL CPU (quad core 2,8GHz)

Temps d'exécution du kernel

Files	Taille Image (pixels)	Kernel CUDA(ms)	Kernel OpenCL (ms)	**1. Ratio CUDA/OpenCL**	Kernel OpenCL CPU (ms)	**2. Ratio CPU/GPU**
PMF_SV	128x115	555,59	588,73	**0,94**		
PMF_SV	909x817	263,19	257,50	**1,02**	897,13	**3,48**
PMF_TV	128x115	134,43	146,18	**0,92**		
PMF_TV	909x817	132,56	147,50	**0,90**	885,90	**6,01**
PMF_FV	128x115	516,93	628,71	**0,82**		
PMF_FV	909x817	310,71	301,44	**1,03**	888,38	**2,95**

True Cumulated View : Portage vers OpenCL

Comparaisons de Bandes passantes = transferts de données Host vers Device.
Carte NVIDIA 2050 – Cuda vs OpenCL



Temps de compilation courts (1ms)
mais parfois très longs (qq centaines ms)
NVIDIA GPU JOCL

BILAN

Les plus

- Réécriture depuis Cuda facile
- Bonnes performances exécution kernel OpenCL vs Cuda
- Bonnes performance OpenCL même sur CPU (via AMD Stream SDK comparable code C OpenMP).
- Exécution sur plusieurs architectures (CPU/GPU) via différentes plateformes (Nvidia/AMD-ATI/Intel) constatée.

Les moins

- Peu de docs, et d'outils de développements.
- Bande passante plus faible sur Nvidia qu'en utilisant Cuda
- Temps de compilation des kernels variables (et parfois longs)

True Cumulated View : Bilan général

Accélérations obtenue importante par rapport implémentation initiale
(jusqu'à x100 sur CPU / x600 sur GPU)

Dilatation en post-traitement
(jusqu'à x36 sur CPU / x25 sur GPU)

CPU
Bon passage à l'échelle – performances stables / données

GPU
Résultats mitigés – performances très dépendantes des données. Facteur d'accélération / CPU insuffisant.

OpenCL
Accélération max en fonction de la configuration hardware.

Perspectives

Evaluation plus poussée d'OpenCL
Automatisation de l'exploitation du matériel disponible
Clusters hétérogènes (CPU/GPU), multi GPUs.

Versions CIVA 2011 stables avec GPU

- Algorithme de reconstruction en Tomographie (ITOC)
- TCV sur OpenCL (OPARUS – Thèse A.Pédron)

Poursuite des travaux

- Nouveaux algorithmes de reconstruction (US/RX)
- Calcul de simulation sur GPU (OPARUS)