

SIMSURF 2

**vers une SIMulation réaliste
des états de SURFace
par calculs massivement parallèles
sur processeurs graphiques**

*Marc JACHYM, Sylvain LAVERNHE,
Christophe TOURNIER, Pierre-Alain BOUCARD*

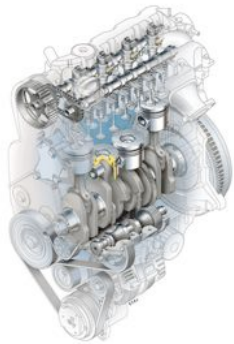
Institut FARMAN –01/12/2014



Contexte et objectifs

Contexte : Simulation des topographies de surface / fonctions techniques

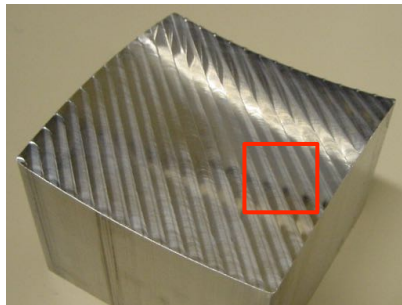
Moteurs (frottement, étanchéité, intégrité de surface)



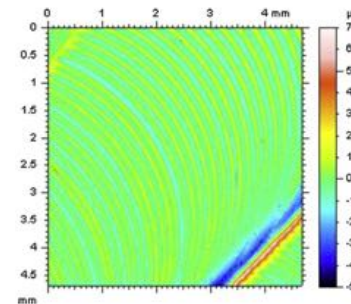
*Prothèse médicale
(frottement, innocuité)*



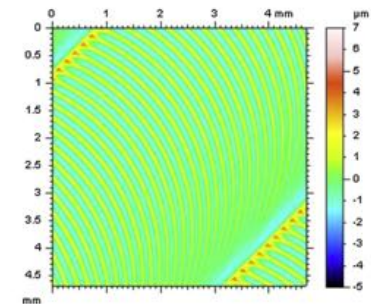
*Verres solaires
(optiques)*



Mesure



Simulation



Ecarts géométriques de l'ordre du μm , temps de simulation ordre de 10 minutes

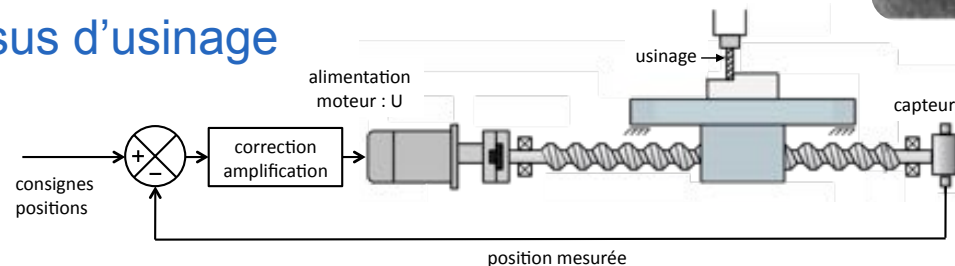
Contexte et objectifs

Besoins : Simulation réaliste des topographies de surfaces usinées par plusieurs opérations successives

➤ Géométrie réelle des outils (dents, usure, bris, etc.)

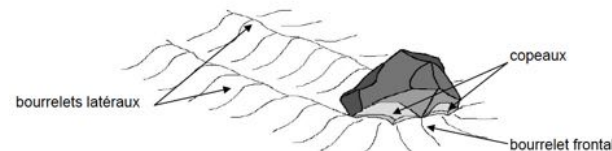
➤ Ecart dus au processus d'usinage

- numériques,
- géométriques,
- mécaniques



➤ Simulation des surfaces fabriquées par des procédés d'abrasion

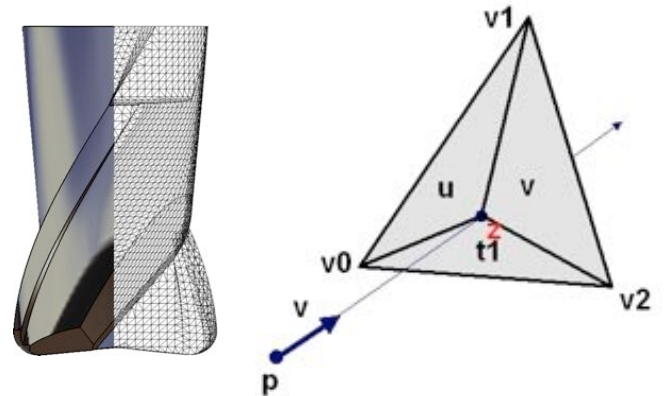
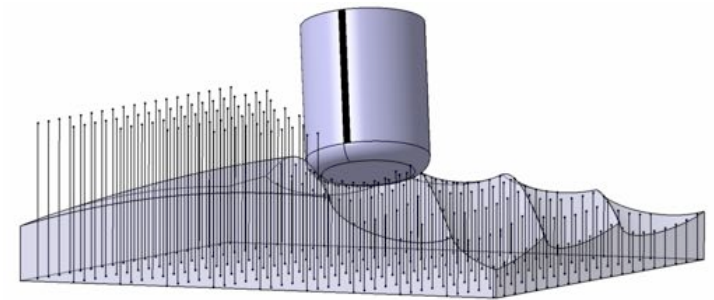
- Chaque grain est un outil, maillage des grains (diamètre 50 μm à 1 μm)



Problématique

Présentation du calcul

- Partition de l'espace en une grille de $N \times M$ droites
 - Outil discrétisé en T triangles
 - Trajectoire avec P positionnements
 - Complexité : $N \times M \times T \times P$
 - Exemple : $N=M=1024$; $T=10^4$; $P = 10^5$
 - Nb opérations $\approx 10^{15}$
 - Occupation mémoire > 10 Go
-
- **Problème adapté au calcul parallèle**
 - **Opération élémentaire relativement simple**

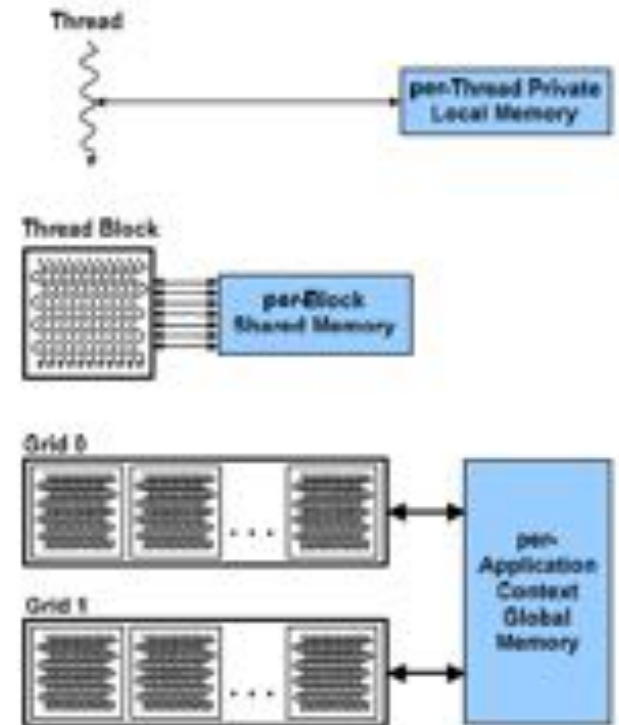


Architecture CUDA

CUDA (Compute Unified Device Architecture)

Architecture logicielle et matérielle qui permet aux cartes graphiques Nvidia d'exécuter des programmes écrits en C, C++ ou autres langages.

- Chaque thread exécute une instance du programme et possède un identifiant
- Chaque thread accède à un espace mémoire réservé réduit mais très rapide
- Les threads peuvent être organisés en blocs et coopèrent entre eux par synchronisation
- Tous les threads d'un même bloc partagent un espace mémoire commun (plus lent)
- Tous les blocs accèdent à la mémoire globale (encore plus lent)



Simulation multi-échelles

Calculs grande échelle ($P=2.10^5$, $T=2.10^3$) (macro géométrie)

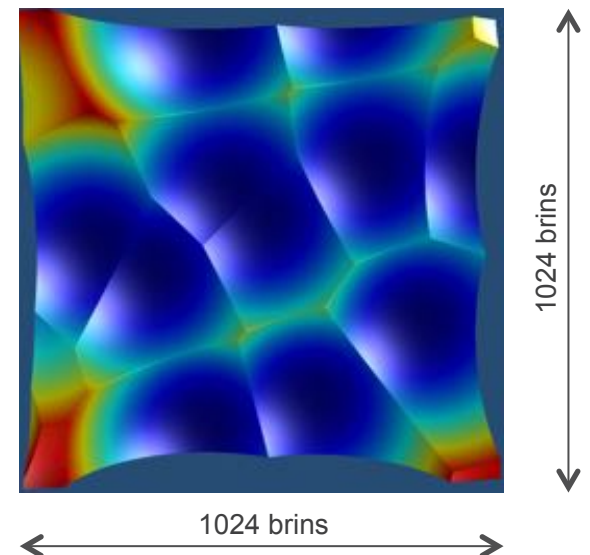
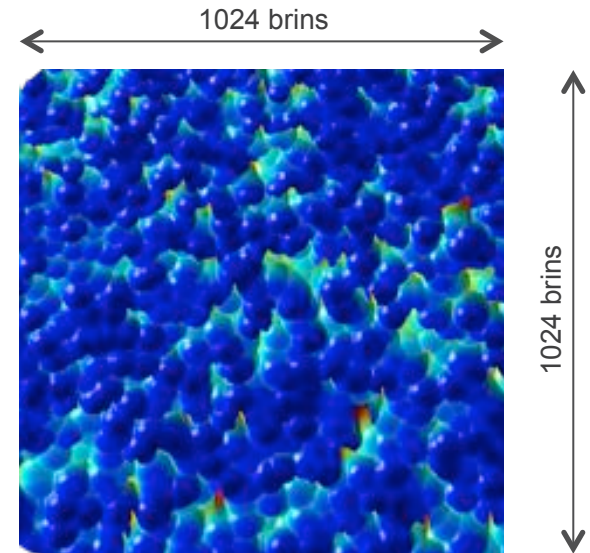
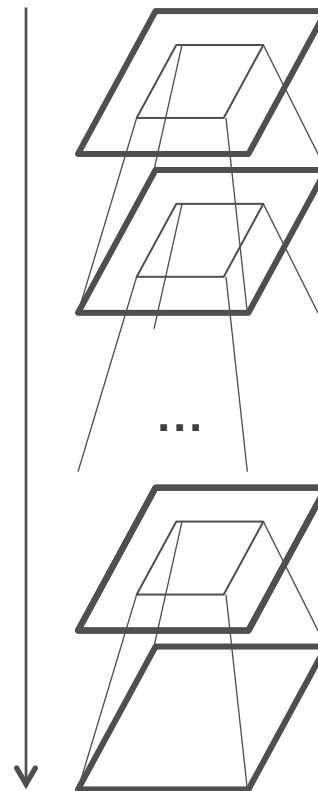
- Thread à granularité élevée
- 1 thread = 1 Triangle * Brins
- Speed-up 64-bit : x5

x64 zoom

Scroll down = x2
(re computation time : 1000 ms)

Calculs petite échelle (micro géométrie)

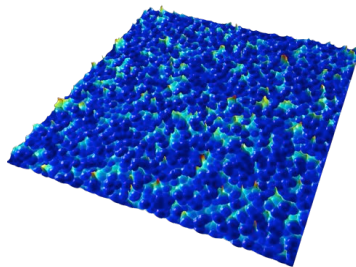
- Thread à granularité élevée
- 1 thread = 1 Brin
- Speed-up 64-bit : x4



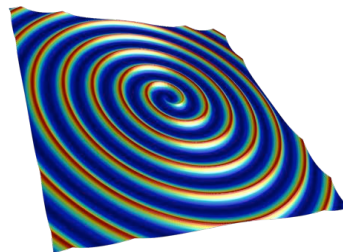
Calculs simple et double précision sur GPU

Amélioration des performances en simple précision

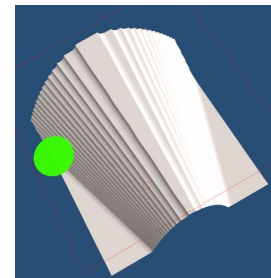
	GeForce GTX 560 Ti		Quadro 4000	
	32-bit	64-bit	32-bit	64-bit
Intel Xeon (6 coeurs)	10,2	2,6	4,4	2,9



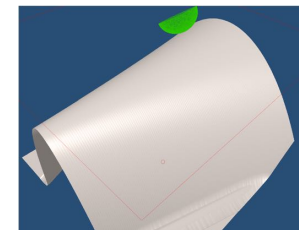
Random



Spiral



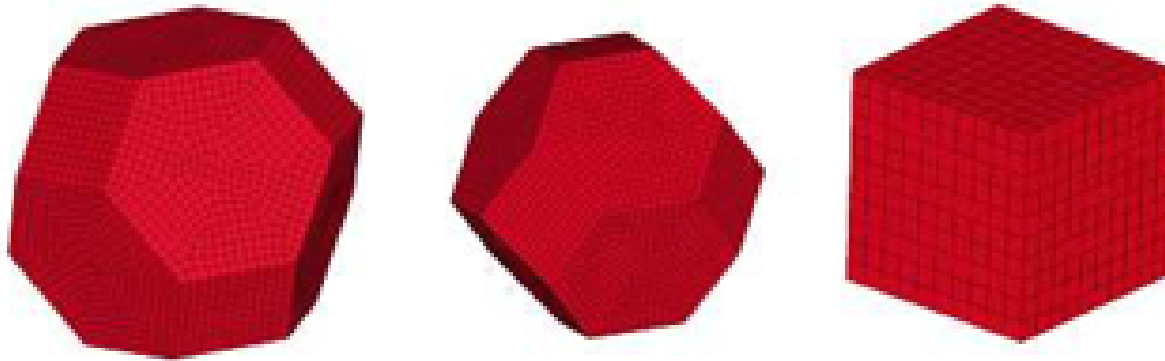
Ebauche



Finition

Objectifs SIMSURF 2

- **Intégrer l'échelle inférieure pour simuler les états de surface produits par abrasion**
 - Modélisation polyédrique des grains ou identification par la mesure
 - Taille du maillage très importante



Représentation polyédrique d'un grain abrasif ^[1]

Papier abrasif P600 :

- taille des grains 25 μm
- taille de maille <2,5 μm

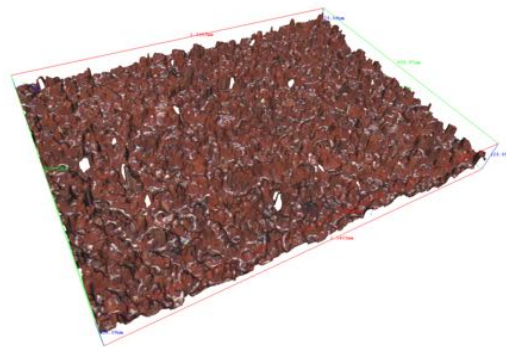
Objectifs SIMSURF 2

➤ Intégrer l'échelle inférieure pour simuler les états de surface produits par abrasion

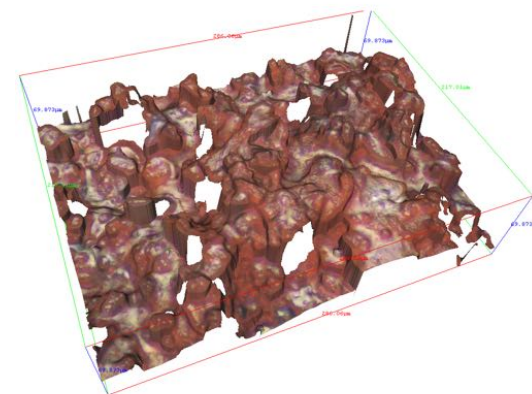
- Modélisation polyédrique des grains ou identification par la mesure
- Taille du maillage très importante



Objectif X5 :
Résolution verticale : 101 nm
Résolution latérale : 14 μm



Objectif X20 :
Résolution verticale : 48 nm
Résolution latérale : 2,9 μm



Objectif X50 :
Résolution verticale : 21 nm
Résolution latérale : 2,13 μm

Mesure d'un papier abrasif P600 (disque $\varnothing 18\text{mm}$)

Objectifs SIMSURF 2

- Implémenter un modèle stochastique pour simuler des pertes de particules abrasives, intégrer leur usure
 - Evolution de l'abrasif au cours du temps
 - Remise en cause des stratégies de parallélisme

**Abrasif
neuf**

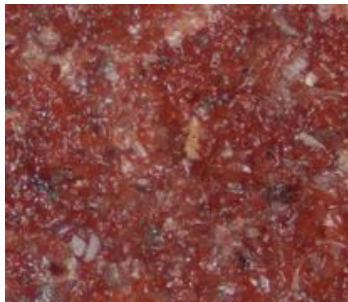
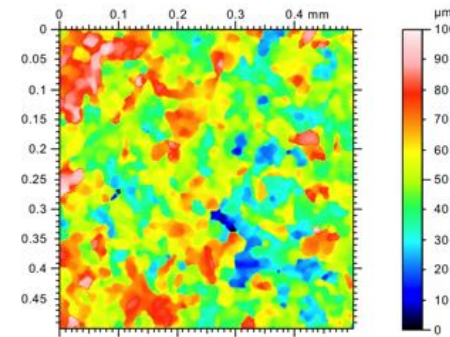
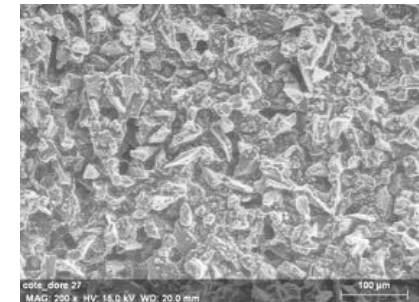


Photo (0,5 mm x 0,5 mm)

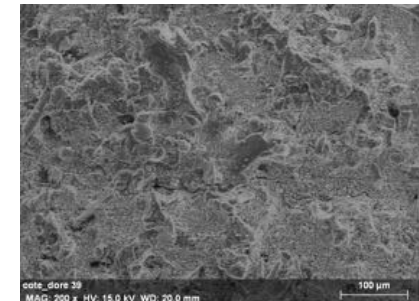
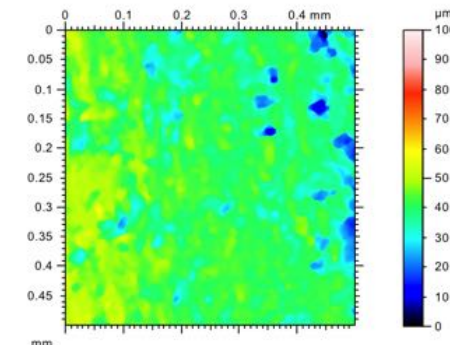
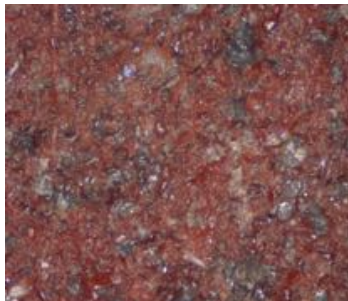


Alicona (0,5 mm x 0,5 mm)



MEB (0,5 mm x 0,5 mm)

**Après
6 minutes**



Objectifs SIMSURF 2

- Diminuer le temps de simulation car taille du problème trop importante : discrétisation trajectoire x modèle outil
 1. Utiliser des clusters de cartes ou des cartes plus performantes
 - GeForce GTX Titan Z (2 cartes GTX Titan)

	Cœurs CUDA	Fréquence	Mémoire	Bande passante	GFLOPS <i>simple précision</i>
GTX 560	384	1,76 GHz	1 Go	128 Go/s	1264
TITAN Z	5760	876 MHz	12 Go	672 Go/s	8122



2. Modifier le moteur de calcul pour utiliser des librairies natives
NVIDIA® OPTIX™ RAY TRACING ENGINE

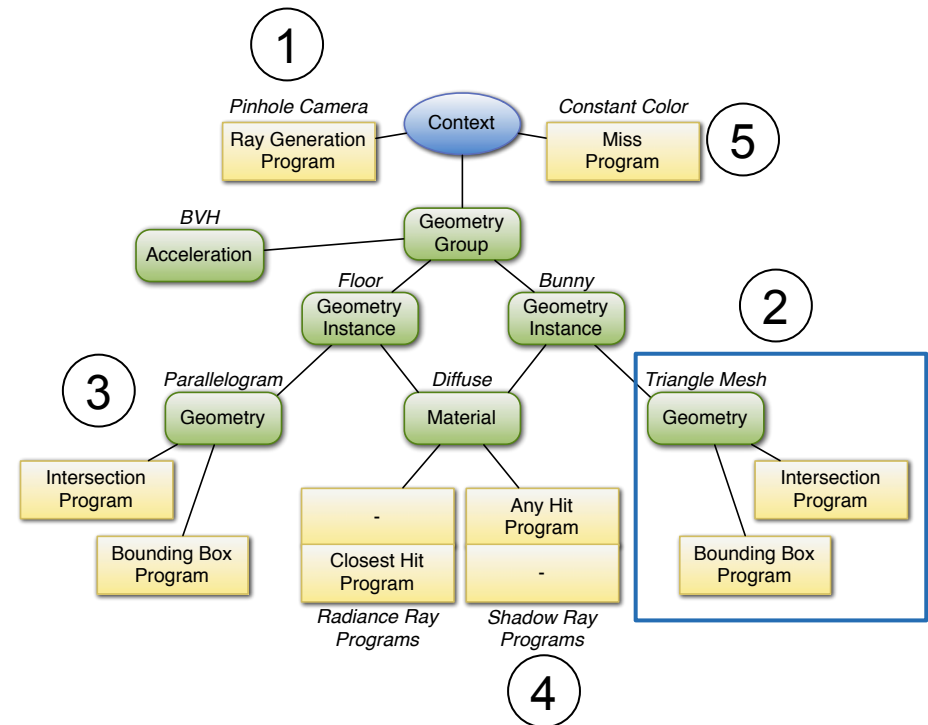
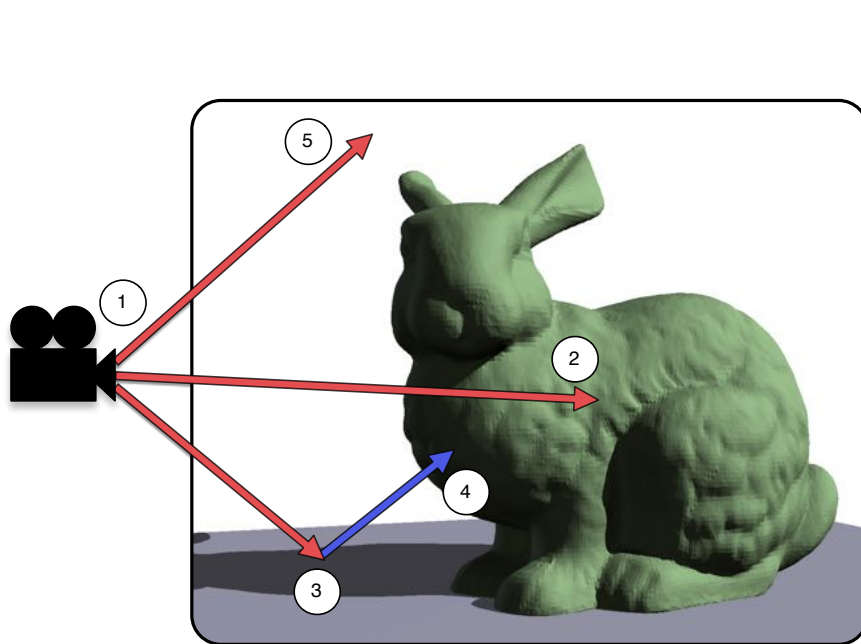
OPTIX™ RAY TRACING ENGINE

Un moteur de Ray Tracing à usage général

- Utilisation de la technique de lancer de rayons pour :
 - rendu, détection de collisions, étude de visibilité ...
- Une infrastructure modulable pour construire des applications à base de lancés de rayons
- Deux parties complémentaires :
 1. Définition d'une structure de données pour le lancé de rayon
 2. Un système de programmation C/Cuda pour construire des rayons, calculer les intersections et traiter ces intersections.



OPTIX™ RAY TRACING ENGINE



L'aspect générique pénalise les performances par rapport à une implémentation manuelle. [2]

➤ Exploiter seulement l'API des intersections avec OptiX Prime

OPTIX™ RAY TRACING ENGINE

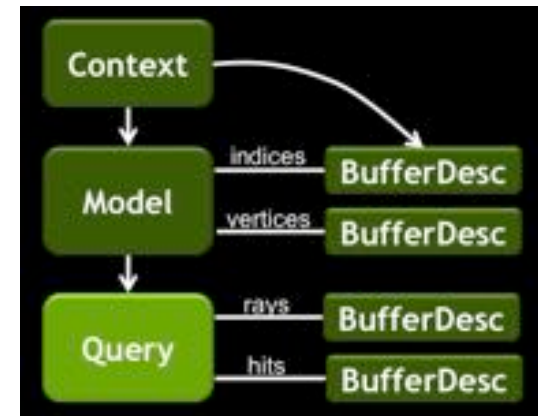
Optix Prime

- « You hand OptiX Prime a list of triangle and rays and it returns the intersections at over 300 million rays per second (on a single GPU), in 5 lines of code»

```
RTP_BUFFER_FORMAT_INDICES_INT3  
RTP_BUFFER_FORMAT_VERTEX_FLOAT3,
```

```
RTP_BUFFER_FORMAT_RAY_ORIGIN_DIRECTION,  
RTP_BUFFER_FORMAT_RAY_ORIGIN_TMIN_DIRECTION_TMAX,
```

```
RTP_BUFFER_FORMAT_HIT_T_TRIID_U_V  
RTP_BUFFER_FORMAT_HIT_T_TRIID
```



Optix : différentes primitives (sphères, plans, triangles, etc.)

Optix Prime : triangles seulement

Conclusions

Encore beaucoup de travail...

1. Amélioration des performances :

- Calcul multi cartes (opérationnel en janvier 2015)
- Exploitation d'OptiX prime (en cours)

2. Modélisation de l'abrasion (Stage de M2 2015) :

- Modélisation géométrique d'un grain
- Modélisation de l'usure
- Simulation de rayage
- Confrontation modèle / réel