

# AAA

# Etudes de cas vidéo/3D

Y.Mathieu

# Plan de la présentation (1)

- Spécificités des applications vidéo et 3D  
Puissances de calcul : quelques ordres de grandeur
- Réorganiser les algorithmes pour faciliter l'accès aux données:
  - L'estimateur de mouvements
  - Le rendu 3D par tuiles

# Plan de la présentation (2)

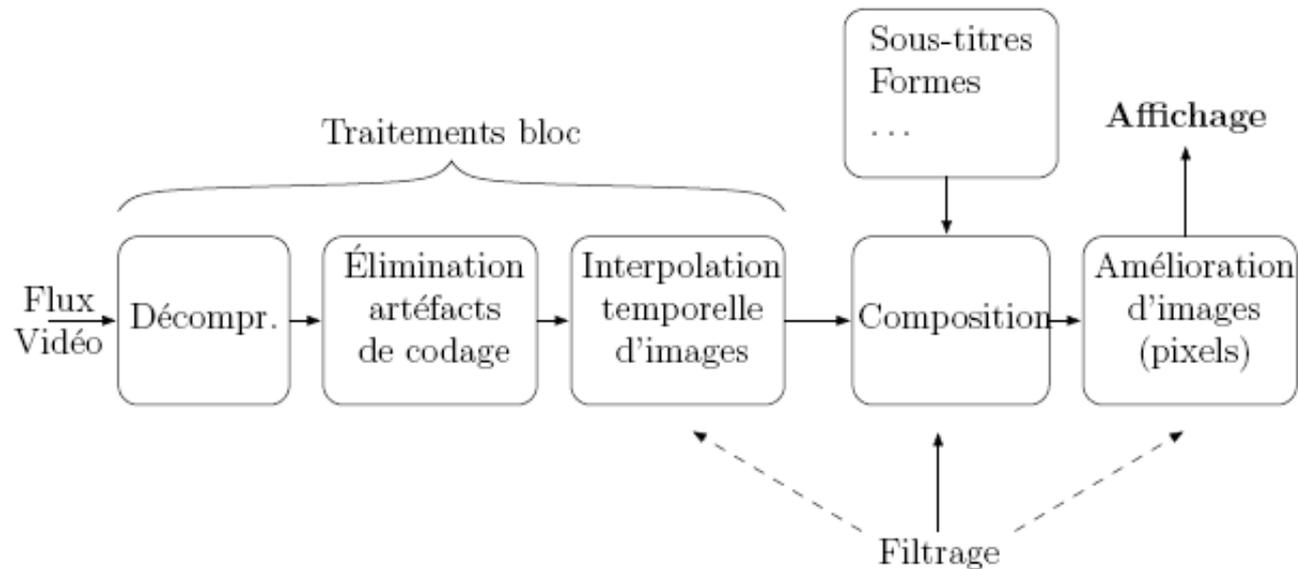
- Gérer la précision des calculs
  - Virgule fixe/Virgule flottante
  - Calcul incrémental
  - Calcul incrémental et rendu par tuile
  - Rendu vidéo : cahier des charges
- Gérer les opérations coûteuses en ressources matérielles
  - Comment éliminer des divisions...

# Plan de la présentation (1)

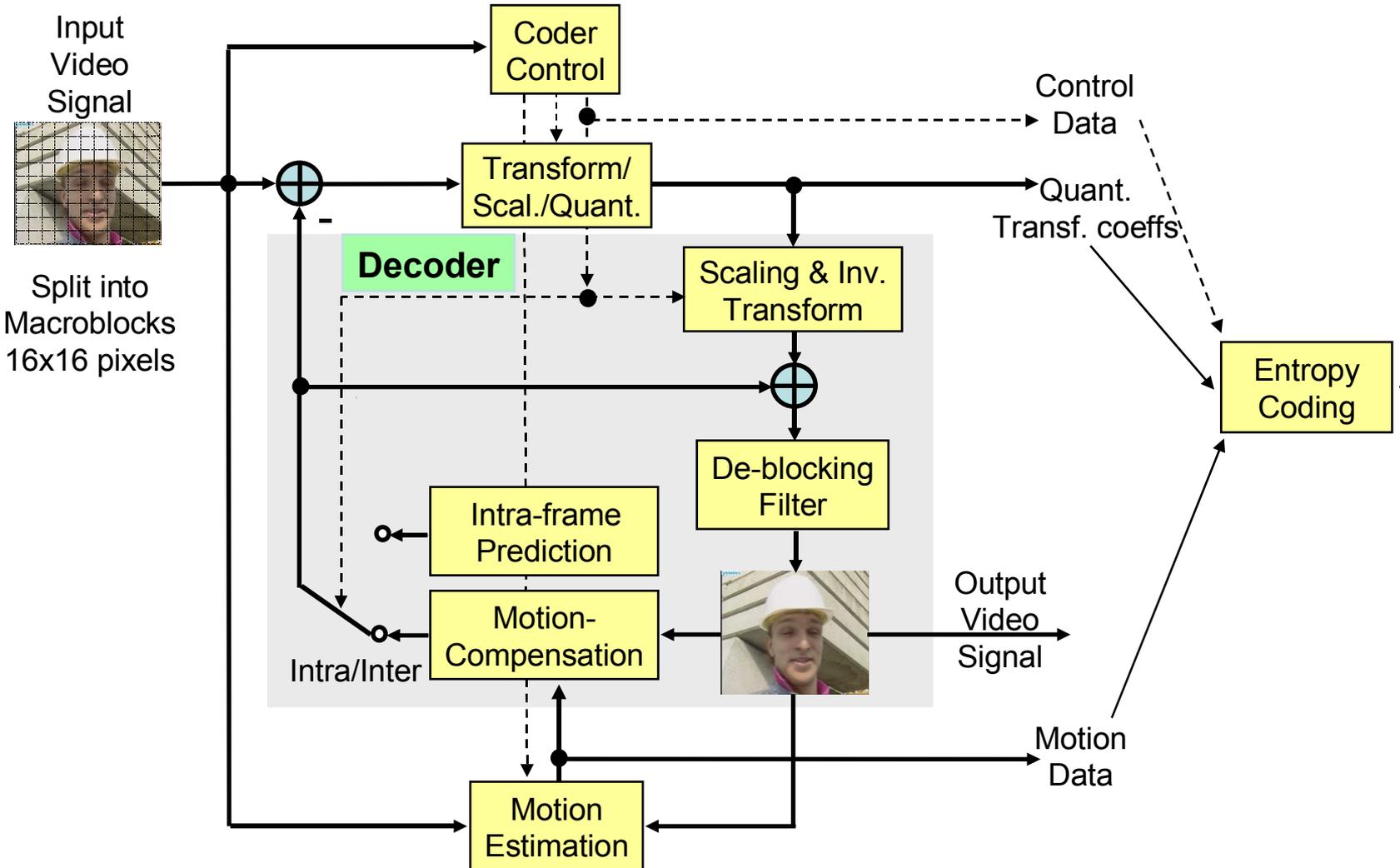
- **Spécificités des applications vidéo et 3D**
- Puissances de calcul : quelques ordres de grandeur
- Réorganiser les algorithmes pour faciliter l'accès aux données:
  - L'estimateur de mouvements
  - Le rendu 3D par tuiles

# Vidéo Numérique

- Opérations principales
  - Codage / Décodage (compression)
  - Filtrage / Améliorations / Rendu



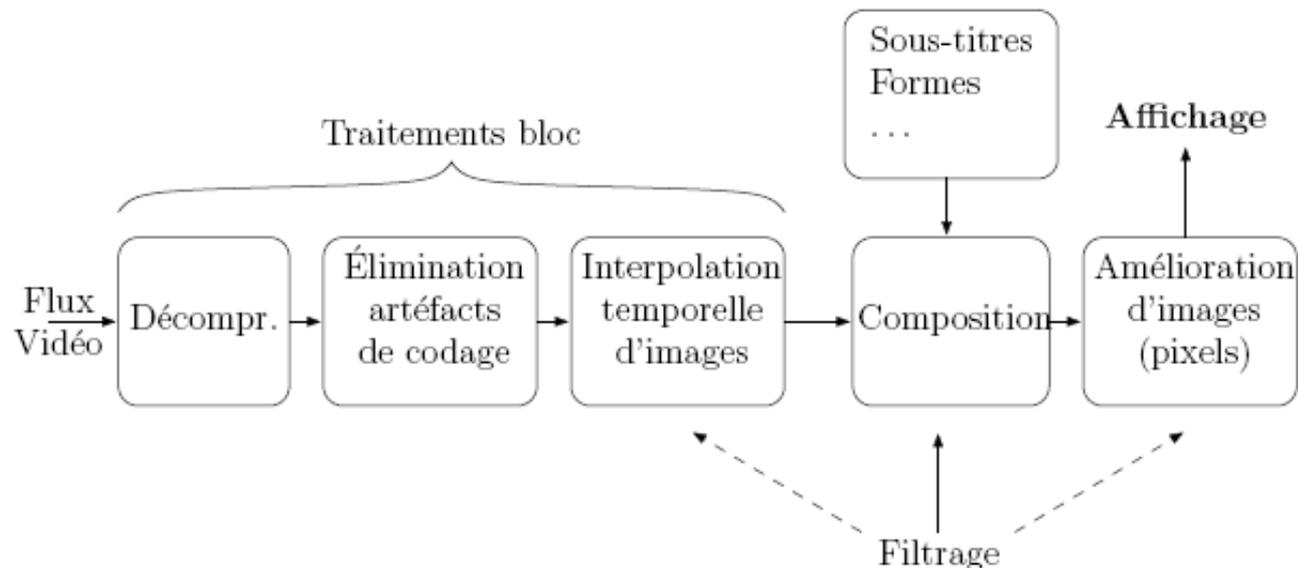
# Vidéo Numérique : Codeur H264



# Vidéo Numérique

- Objets traités

- Images vidéo : rythme régulier temps réelle 25im/s
- Hiérarchie simple : image / blocs / pixels



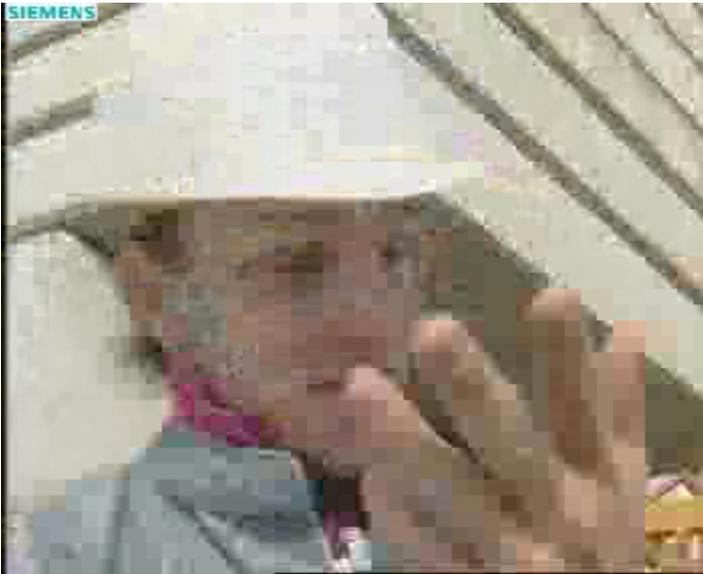
# Vidéo Numérique

- Exemples de filtrages (<http://www.compression.ru/video>)
  - Suppression de bruit (pour faciliter la compression)



# Vidéo Numérique

- Exemples de filtrages (<http://www.compression.ru/video>)
  - « Deblocking filter »



# Vidéo Numérique

- Exemples de filtrages (<http://www.compression.ru/video>)
  - « Smart Sharpen filter »



# L'estimation de mouvement



Image différence sans  
compensation de mouvement



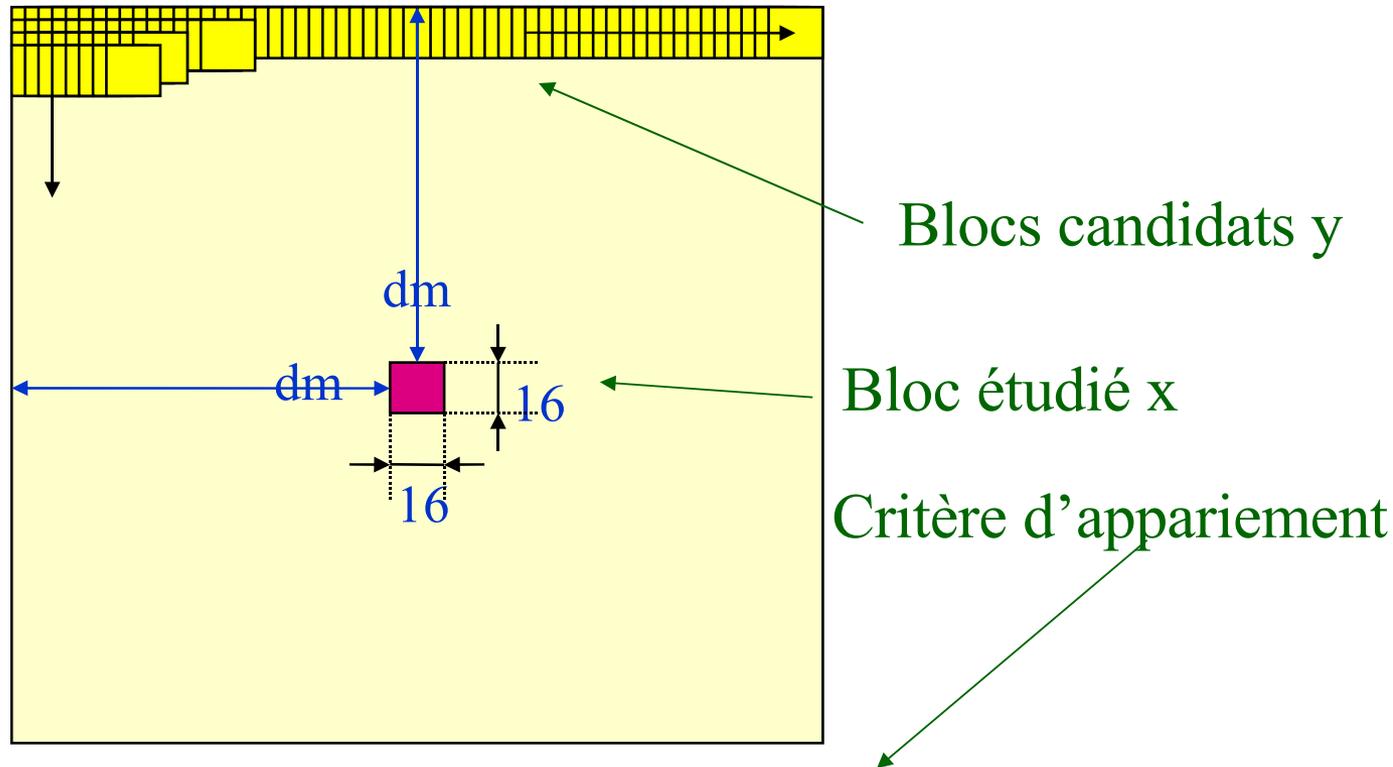
Image différence avec  
compensation de mouvement

# L'estimation de mouvement

- Uniquement translations de blocs dans l'image



# L'estimation de mouvement

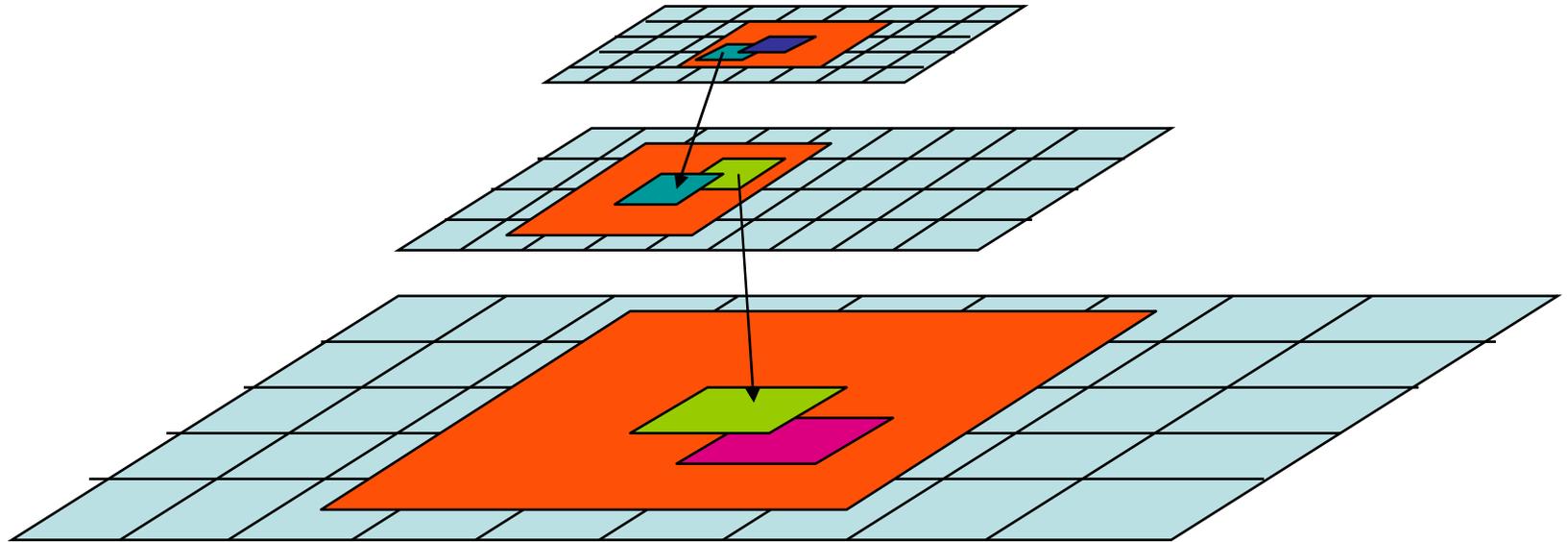


$$MAD = \sum_{l=1 \text{ à } 16} \sum_{k=1 \text{ à } 16} |x_{l,k} - y_{l+i,j+k}|$$

$$P_c \sim (2 \cdot w + 1)^2 \cdot F_{\text{pix}} / 256$$

De nombreux algorithmes sous-optimaux

# L'estimation de mouvement



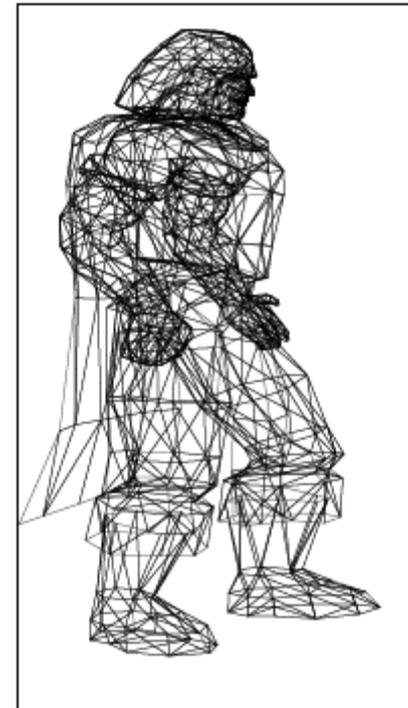
- un ou plusieurs prédicteurs à chaque résolution
- $w_i$  : excursion au niveau  $i$  ;  $r_i$  résolution au niveau  $i$
- $p_i$  : nombre de prédicteurs utilisés au niveau  $i$
- $W = \sum(w_i/r_i)$  : excursion totale

$$P_c \sim \sum(r_i \cdot p_i \cdot (2 \cdot w_i + 1)^2 \cdot F_{\text{pix}} / 256)$$

# Qu'est ce que la 3D ?

- Une image 2D
- Une image synthétique (créée entièrement par ordinateur)
- Une impression de profondeur
  - Des effets d'éclairage de la scène
  - Des effets de perspectives
  - Des « textures photo-réalistes »
- Des animations « réalistes »

# Le rendu d'une scène graphique 3D



Copie d'écran du jeu « Vampire » et  
Détail du modèle maillé d'un personnage

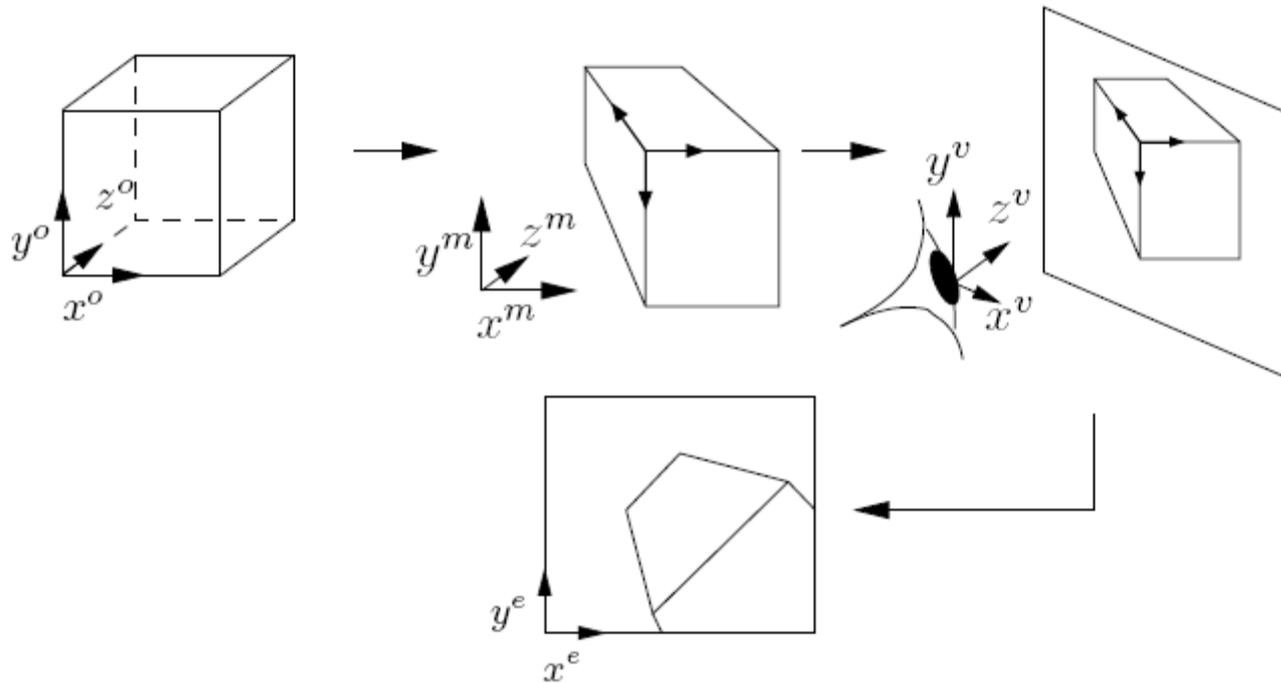
# Modélisation d'une scène

- Un « objet » est « placé » dans la scène
- Un objet est composé de primitives géométriques de base : le triangle
- Un triangle est défini par la position de ses trois sommets dans la scène 3D.

- Un sommet :  $S_0 = \begin{pmatrix} x_0^o \\ y_0^o \\ z_0^o \end{pmatrix}$



# Le pipeline de transformations géométriques



# Les transformations géométriques

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

- Transformations (rotations, symétries, homothéties) : produit matriciel

- La translation : addition vectorielle

- Utilisation de coordonnées homogènes (majuscules)

- Espace 4D avec  $w$  coordonnée homogène

- Un point 3D

- Un vecteur 3D

- Coordonnées 3D du point S.

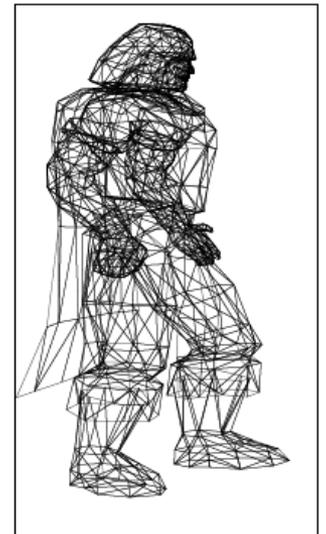
$$\begin{pmatrix} X & Y & Z & w \end{pmatrix}^t$$

$w \neq 0$

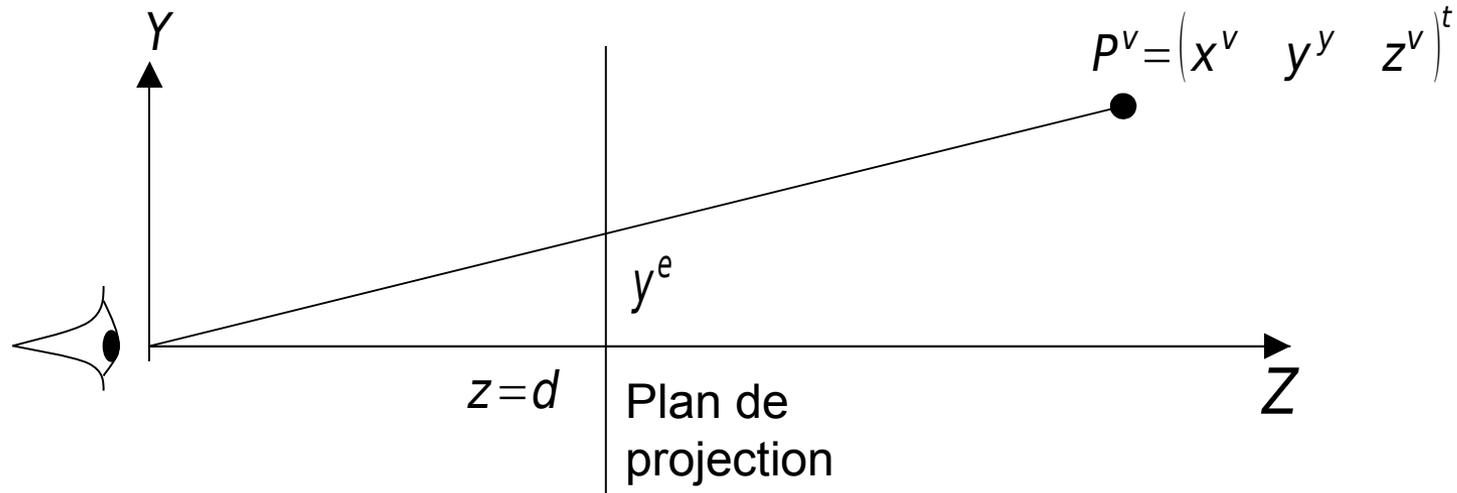
$w = 0$

$$\begin{pmatrix} X/w & Y/w & Z/w \end{pmatrix}^t$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



# La transformation en perspective

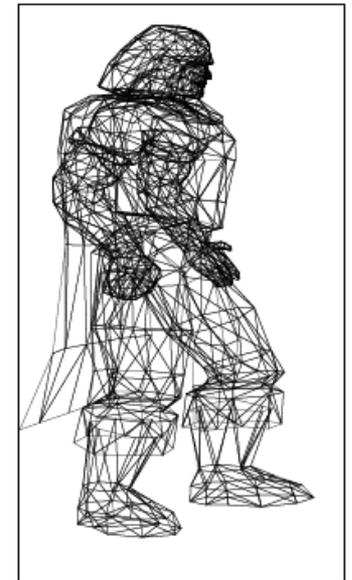


$$\frac{y^e}{y^v} = \frac{d}{z^v}$$

$$\begin{pmatrix} X^e \\ Y^e \\ Z^e \\ w^e \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} X^v \\ Y^v \\ Z^v \\ 1 \end{pmatrix}$$

# Transformation géométrique : Bilan

- Modélisations des mouvements hiérarchiques : produits de matrices  $4 \times 4$  en cascades + une matrice de projection
- Reste une seule matrice globale :
  - 16 multiplications
  - 12 additions
  - 2 divisions (1 inversion, 2 multiplications)
- Par point définissant une primitive.



# Parcours des géométries

- Parcours 2D (espace écran)
- Passages des primitives aux pixels
- Détermine exactement les pixels recouverts
  - Parcours pixel après pixel  
(*INCREMENTAL*)
  - Calcul des effets de lumières
  - Placage de texture
  - Elimination des surfaces cachées

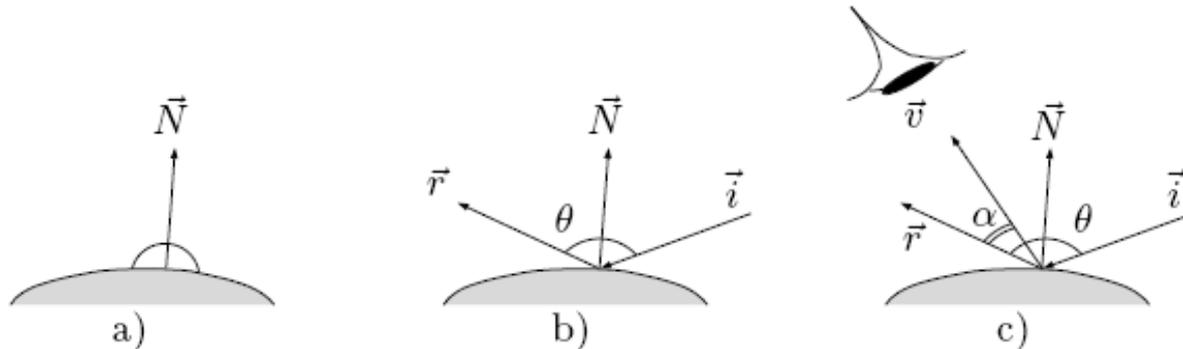
# Eclairage de la scène

- Les sources de lumière éclairent la scène
- Trois types de sources lumineuses

• Lumière ambiante :  $I = k_a I_a$

• Lumière diffuse, Loi de Lambert pour les objets mats :  $I = k_d I_d \cos \theta$

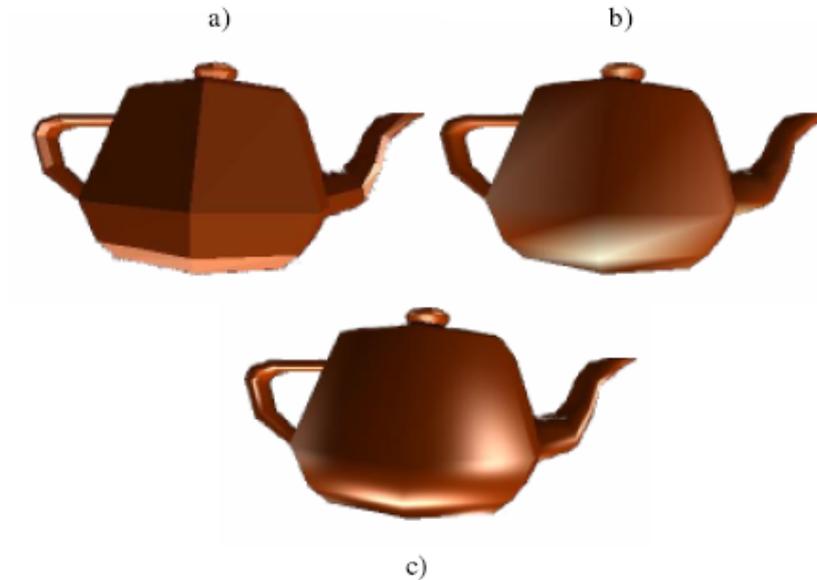
• Lumière spéculaire (rugosité) :  $I = k_s I_s \cos^n \alpha$



- Deux produits scalaires par source d'éclairage pour chaque point définissant la primitive.

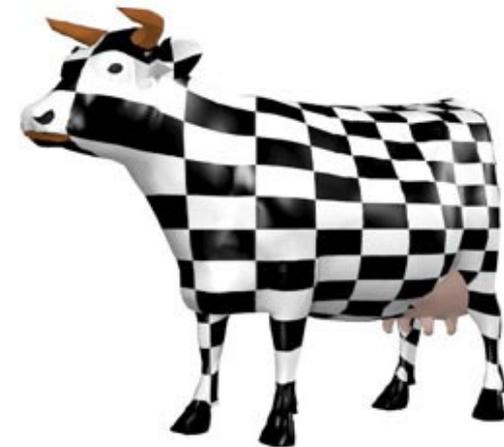
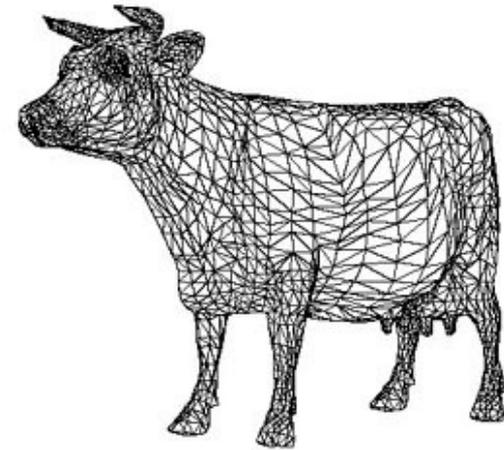
# Parcours des géométries

- Les effets de lumières, méthodes d'interpolation (Gouraud, Phong...)



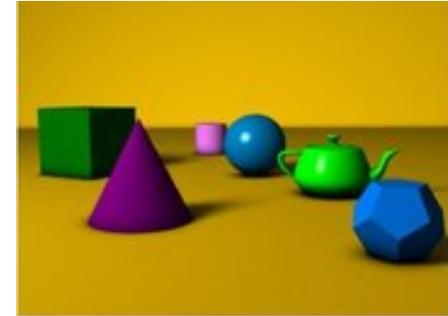
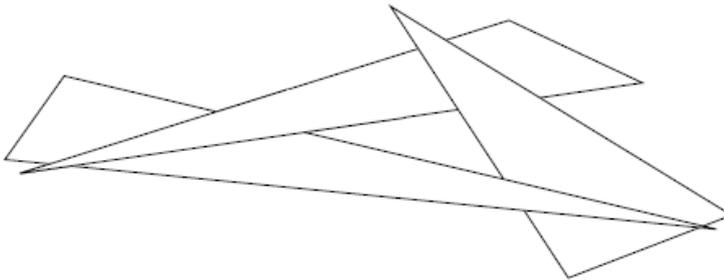
# Parcours des géométries

- Placage de texture
- Une texture est associée à un objet
  - Correspondance entre les coordonnées objet  $(x^0, Y^0, z^0)$  et texture  $(u, v)$  (transformation affine)
  - Dépendance non linéaire  $(u, v)$  avec  $(x^e, y^e)$
  - Dépendance linéaire de  $(u/z, v/z)$  avec  $(x^e, y^e)$
- Filtrage simpliste des textures
- Textures fixes



# Parcours des géométries

- Elimination des surfaces cachées
- Liste de priorité : 2,5D, ne fonctionne pas toujours
- Z-buffer : mémoire de profondeur pour chaque pixel

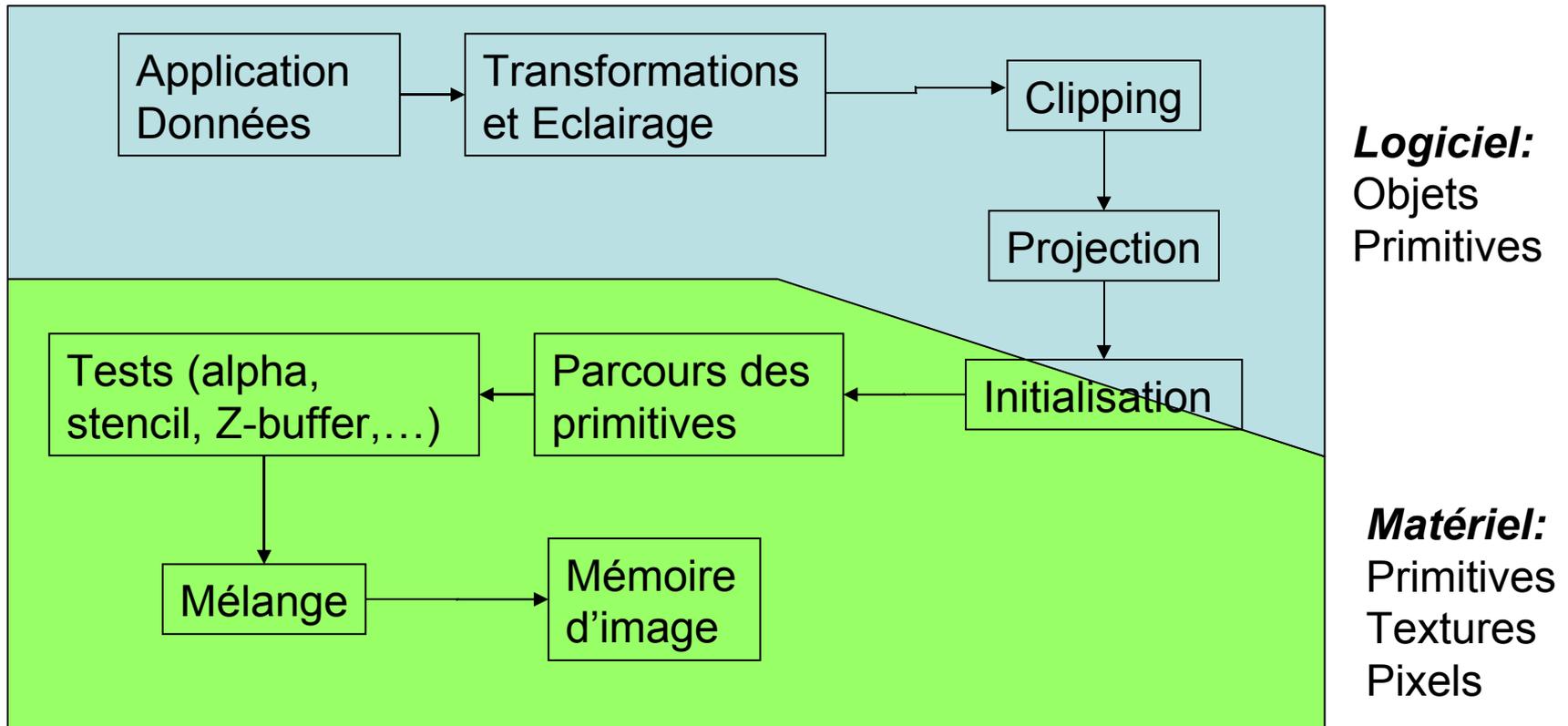


Une scène 3d simple



Le Tampon de profondeur

# Le pipeline graphique



# Plan de la présentation (1)

- Spécificités des applications vidéo et 3D
- Puissances de calcul : quelques ordres de grandeur
- Réorganiser les algorithmes pour faciliter l'accès aux données:
  - L'estimateur de mouvements
  - Le rendu 3D par tuiles

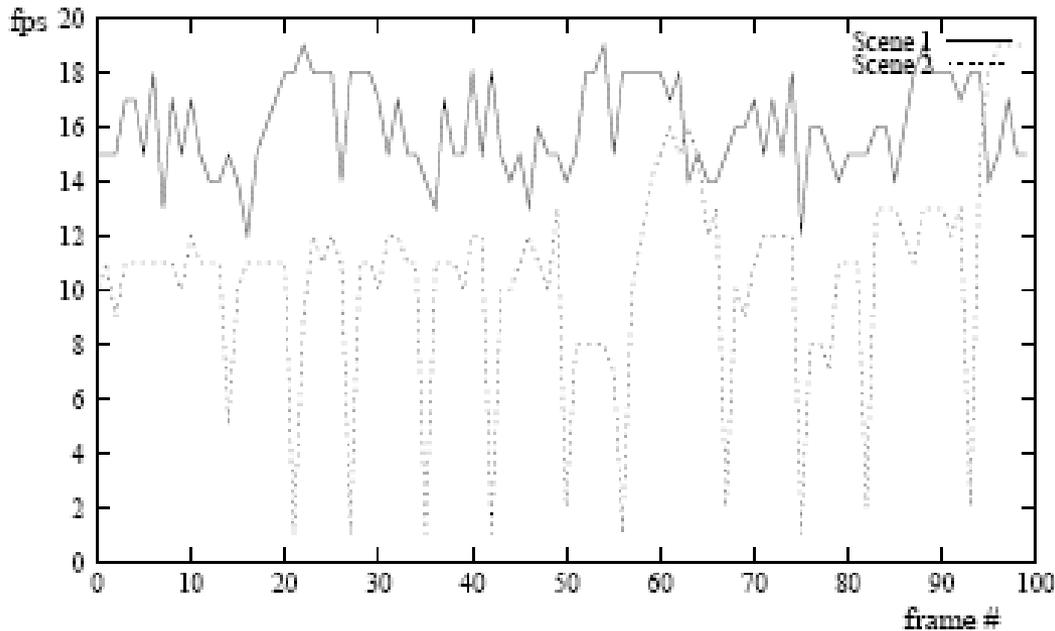
# Video : Estimation de mouvement

- Télévision numérique : 13,5 Mpel/s
- Opération de base VAD (valeur absolue de différence de 2 pixels (complexité : 2 additions 8 bits)
- MPEG2 : 2 images de référence (voir interpolation)
- Taille des images : 720x576
- Excursion raisonnable : -128/+127 (reportages sportifs)
- Puissance de calcul nécessaire : 6,97 Gadd/s
- H264 :
  - Trames de référence multiples 8..16..
  - Blocs 16x16, 8x8, 4x4.....
  - Prédiction trame, image...

Quelles conclusions peut on en tirer ?

# Jeux video-3D « Frame Rate »

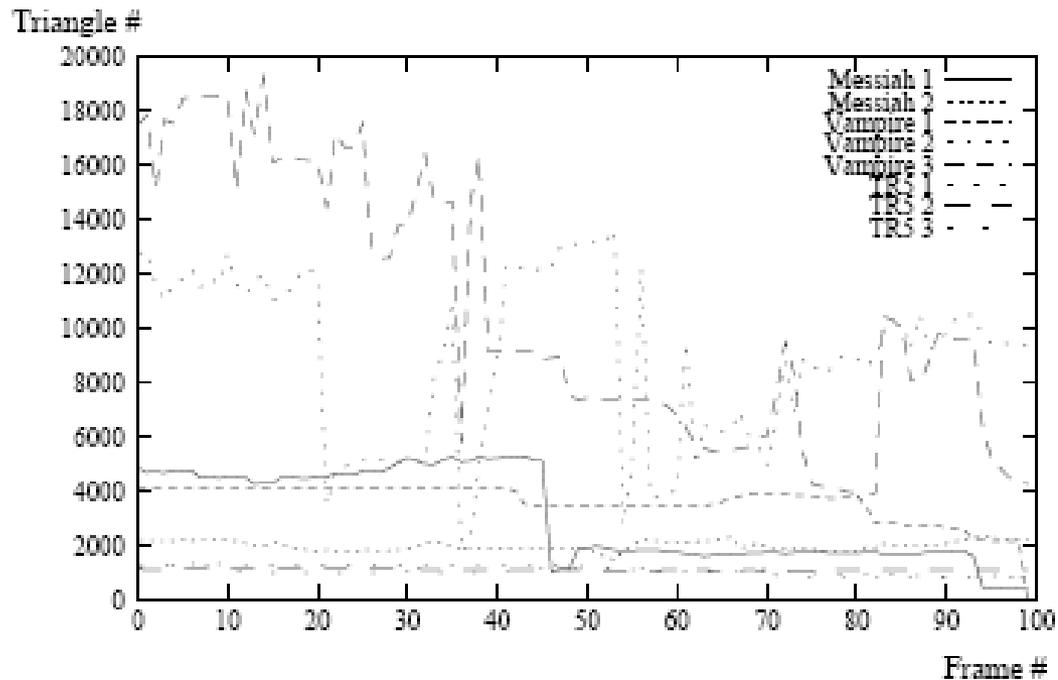
- Objectif : dimensionner un système de traitement 3D.
- Méthode : définir un modèle statistique des scènes



La vitesse de rendu des scènes 3D est variable et dépend de leur complexité. Une vitesse moyenne de 20 images/sec avec un minimum de 10 images/sec acceptable pour la plupart des jeux.

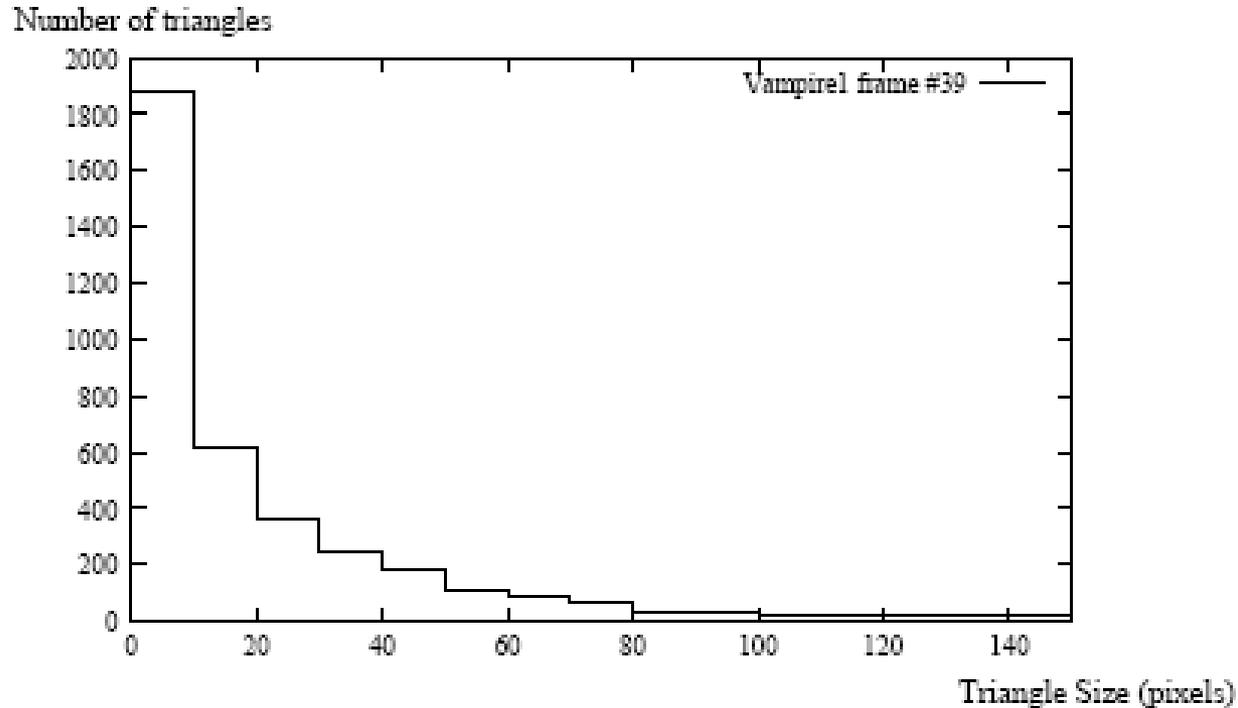
Question : quelle est la conséquence sur les architectures de traitement ?

# Jeux video : complexité de la scène



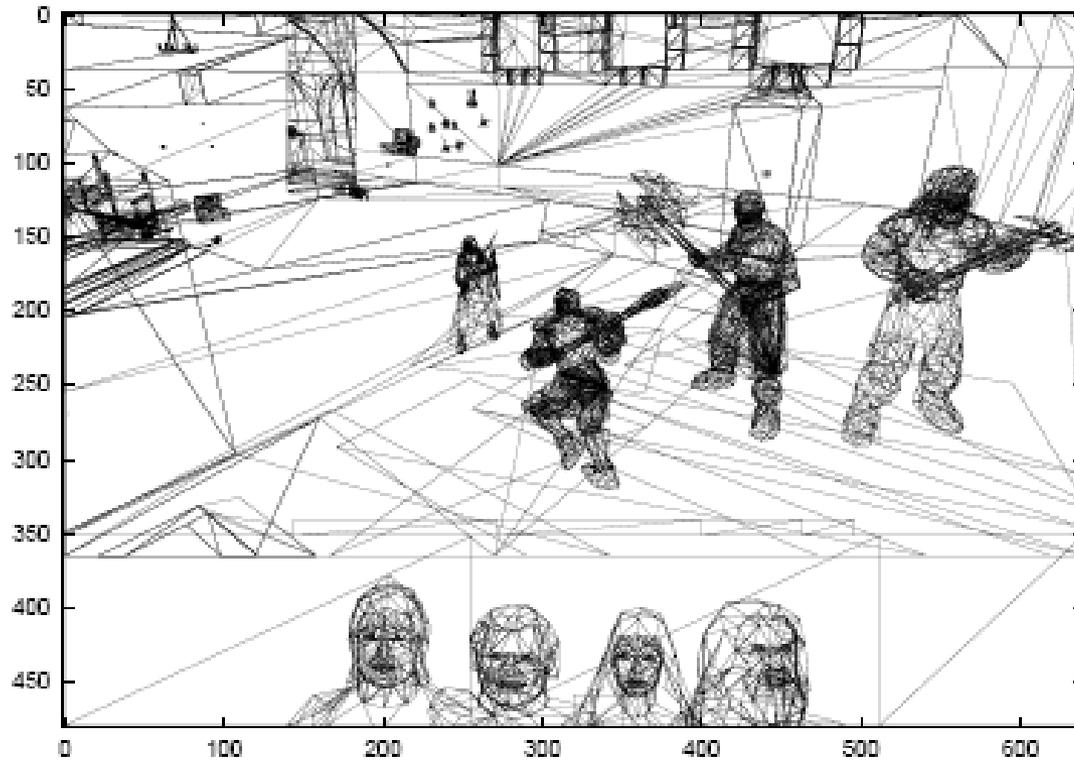
Une complexité de 15000 triangles par scène semble une moyenne raisonnable dans les conditions précédentes.

# Statistiques de taille des triangles



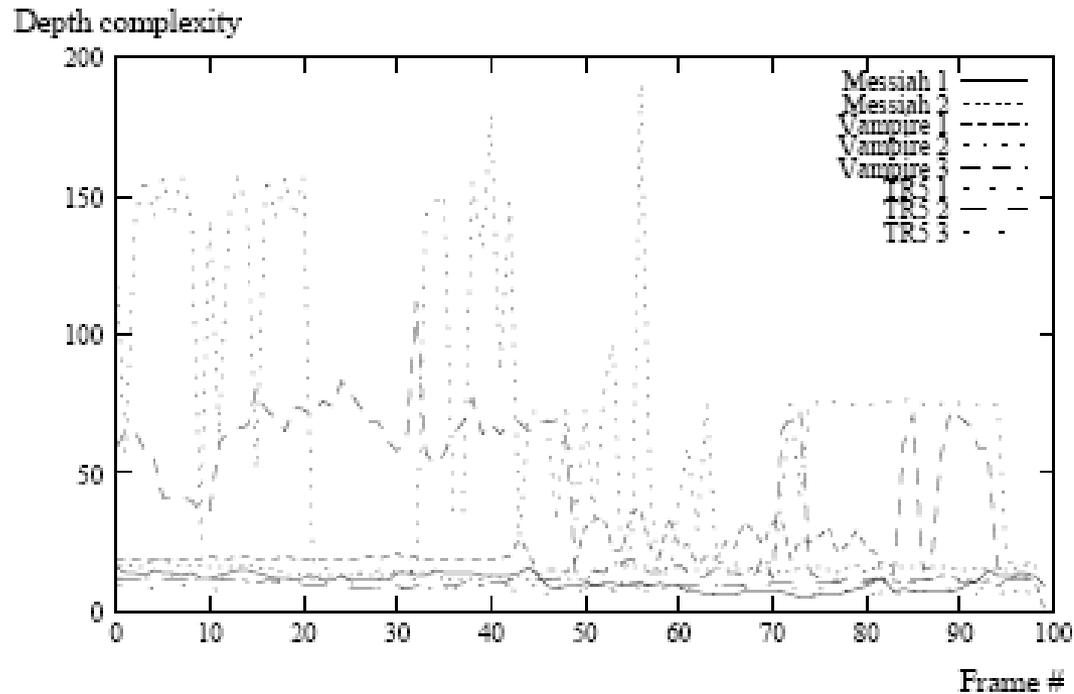
Question : quelle est la conséquence de la variabilité de tailles des triangles ?

# Statistiques de taille des triangles



Question : quelle est la conséquence de la variabilité de tailles des triangles ?

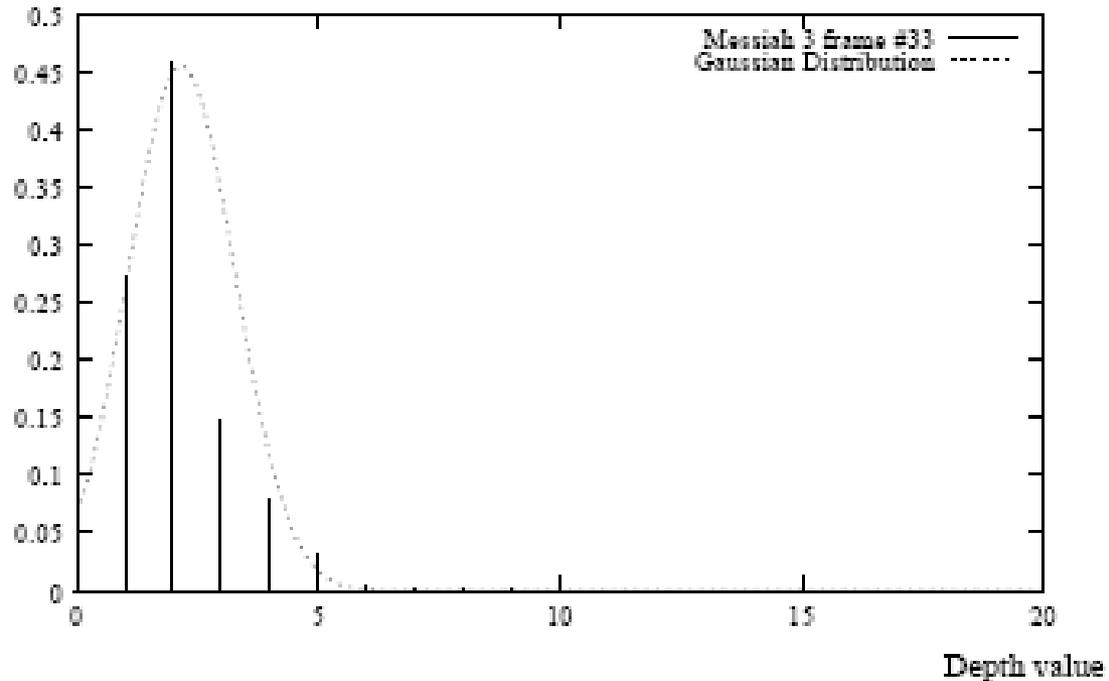
# Statistiques de profondeur (Z-buffer)



Dans les exemples proposés : maximum : 150, moyenne 2,5 , median 3

# Statistiques de profondeur (Z-buffer)

Normalized pixel number



Distribution typique : Nombre de pixels =  $f(\text{profondeur})$

# 3D : Applications numérique

*TRAVAUX DIRIGES : A vous de travailler !!!*

Il s'agit d'estimer la puissance de calcul nécessaire (en opérations élémentaires/s pour un rendu matériel 3D ayant les caractéristiques suivantes :

- L'accélérateur matériel prends en charge la transformation géométrique (transp 12)
- Scène de 16383 triangles avec la statistique suivante :
  - 8192 triangles de 16 pixels
  - 4096 triangles de 32 pixels
  - ..
  - 1 triangle de 128K pixels
- L'image 512x512 doit être générée en 40ms

# 3D : Applications numérique

- Pas d'éclairage à calculer mais placage de texture avec interpolation bilinéaire.
- Profondeur moyenne du Z-buffer : 3.
- Pixels sur 3 composantes R,G et B.
- Quelle méthode utiliser ?
  - Déterminer les opérations nécessaires au parcours d'un triangle
  - Multiplier par le nombre de triangles

# Plan de la présentation (1)

- Spécificités des applications vidéo et 3D
- Puissances de calcul : quelques ordres de grandeur
- Réorganiser les algorithmes pour faciliter l'accès aux données:
  - L'estimateur de mouvements
  - Le rendu 3D par tuiles

# Définition du problème

- Obtenir une puissance de calcul importante n'est plus un problème insurmontable
- L'accès des accélérateurs matériels aux données est devenu le goulot d'étranglement principal.
- Ces deux aspects doivent être étudiés conjointement pour obtenir un résultat « équilibré »
- Dans le cas du « temps réel » (vidéo) les systèmes de caches statistiques ne sont pas viables (profils Mpeg)

# Estimateur de mouvement

- *A vous (ENCORE !!!) de travailler.*
- Hypothèses : blocs 16x16, Fenêtre de recherche (-32/+31), Image 512x512, Débit 10Mpels/sec
- Fréquence de fonctionnement d'un opérateur de VAD : 500 Mhz.
- Fréquence de fonctionnement des mémoires internes : 32Kx8 250 Mhz, 256x8 500 Mhz

# Estimateur de mouvement

- L'algorithme peut être réécrit de la façon suivante :

```

errmin = max ;
dxref = 0 ; dyref = 0 ;
for (i=-32;dx <32;dx++)
    for(dy=-32;dy<32;dy++) {
        err = 0 ;
        for(i=0;i<16;i++)
            for(j=0;j<16;j++)
                err = err + abs(pix(i,j)-ref(i+dx,j+dy)) ;
        if(err < errmin) {
            errmin = err ;
            dxref = dx ;
            dyref = dy ;
        }
    }
}

```

# Estimateur de mouvement

- *1/ Déterminer la puissance de calcul nécessaire (VAD/sec) ainsi que le nombre d'opérateurs (parallélisme)*
- *2/ Réécrire l'algorithme en faisant apparaître explicitement ces opérateurs (déroulement des boucles)*

# Estimateur de mouvement

- L'algorithme d'estimation de mouvement peut être réexprimé en « pseudo langage » sous la forme suivante :

```
errmin = max ;  
dxref = 0 ; dyref = 0 ;  
for(i=0;i<16;i++)  
  for(j=0;j<16;j++)  
    for (dx=-32;dx <32;dx++)  
      for(dy=-32;dy<32;dy++)  
        err[dx,dy] = err[dx,dy] + abs(pix(i,j)-ref(i+dx,j+dy)) ;  
for (dx=-32;dx<32;dx++)  
  for(dy=-32;dy<32;dy++) {  
    if(err[dx,dy] < errmin)  
      errmin = err[dx,dy] ;  
      dxref = dx ;  
      dyref = dy ;  
  }
```

# Estimateur de mouvement

- *3/ Réécrire l'algorithme 2 en faisant apparaître les opérateurs (déroulement des boucles).*
- *4/ Déterminer les ressources de calcul exactes nécessaires à l'exécution des algorithmes (cas 1 , cas 2)*
- *5/ Déterminer les ressources de mémorisations nécessaires à l'exécution de l'algorithme suivant les deux solutions.(hors acheminement des données)*

# Estimateur de mouvement

- *On utilise maintenant une mémoire interne et un parcours régulier (ligne de blocs, bloc par bloc dans chaque ligne)*
- *Pour chacun des cas 1 et 2 :*
- *3/ Déterminer des organisations de mémoire interne permettant de réduire le débit nécessaire avec la mémoire externe*
- **4/ Déterminer la taille de ces mémoires et les débits associés suivant la taille considérée**



# Plan de la présentation (1)

- Spécificités des applications vidéo et 3D
- Puissances de calcul : quelques ordres de grandeur
- Réorganiser les algorithmes pour faciliter l'accès aux données:
  - L'estimateur de mouvements
  - Le rendu 3D par tuiles

# Rendu 3D par tuile

- L'objectif est une fois encore exploiter la localité des traitements pour éviter des opérations d'entrées sorties inutiles.
- L'image à rendre est découpée en tuiles de tailles  $N_t \times N_t$
- Toute les opérations de rendu sont effectuées au niveau des tuiles

# Rendu par tuile

- *Avantages :*
  - Suppression du Z-buffer externe (limité à une mémoire interne au processeur) et aux E/S associées
  - Anticrénelage de qualité facilité (voir suite)
  - De manière générale, toutes les traitements nécessitant des E/S intensives sont favorisés.

# Anticrénelage

- Filtrage des bords des primitives.
- Solution : rendu à plus grande résolution, puis moyenne des contributions (ex : 2x2)
- Nécessite une mémoire d'image (de tuile !!!) 4 fois plus grande



# Rendu par tuile

- *Inconvénients :*
  - *Algorithme naïf à proscrire :*
    - *Pour chaque tuile...*
      - *Pour chaque triangle...*
      - *Pour chaque pixel...*
  - Nécessite un tri des primitives interceptant la tuile.
  - Trouver le bon compromis de taille de tuile

# Rendu par tuile

***Utilisation du rendu différé : Dans le cas du rendu / tuile, on peut différer le rendu en stockant les adresses plutôt que des données. Le rendu proprement dit ne sera effectué q'une seule fois.***

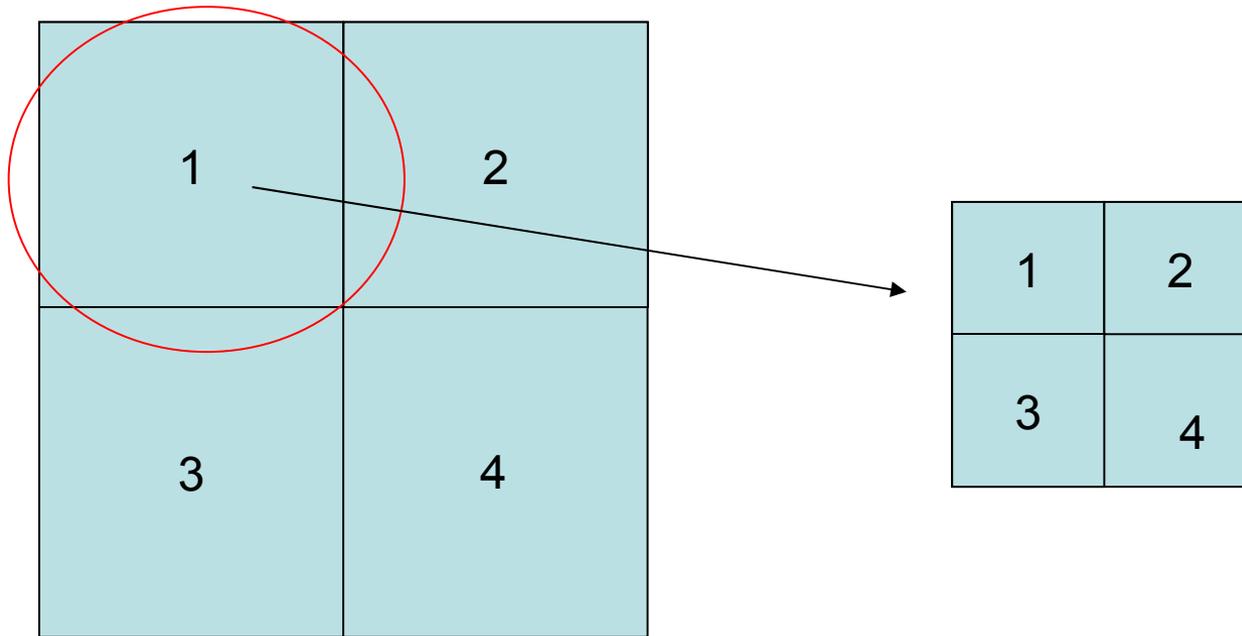
- ***Avantage : nouveau gain en bande passante (objets)***

# Rendu par tuile

## *Limitation des parcours par un rendu en Quadtree*

- *Eviter le parcours de pixels déjà couverts*
- *Maintien d'une liste des pixels couverts à chaque niveau hiérarchique*
- *Utilisable en 2,5 D*
- *Algorithme : du plus proche vers le plus éloigné.*

# Parcours Quadtree



# Exercice

- Déterminer la structure d'un compteur permettant d'effectuer le parcours d'un quadtree pixel par pixel.
- Déterminer les aménagements nécessaires pour permettre des sauts d'adresse lorsqu'un niveau hiérarchique est plein

# Exercice

- Déterminer une structure matérielle implémentant le quadtree (tuille 16x16)
- Déterminer une structure matérielle permettant la mise à jour des informations de couverture à chaque niveau hiérarchique

# Plan de la présentation (2)

- **Gérer la précision des calculs**
  - Calcul incrémental
  - Calcul incrémental et rendu par tuile
  - Virgule fixe/Virgule flottante
  - Rendu vidéo : cahier des charges
- **Gérer les opérations coûteuses en ressources matérielles**
  - Comment éliminer des divisions...

# Calcul Incrémental

- Exploiter le parcours systématique d'un espace de données pour déterminer la valeurs de fonctions polynomiales de façon simple:
- $U[x,y] = a*x+b*y+c$  (2 multiplications par calcul)
- $U[0,0] = c$
- $U[x+1,y] = U[x,y]+a$  (une addition par déplacement)

# Calcul Incrémental

- Pas ou peu utilisable pour des calculs non linéaires
- Exemple : transformée perspective
  - $U[x,y] = (a*x+b*y+c)/(d*x+e*y+f)$
  - Avec  $z = d*x+e*y+f$
- Parcours à  $z$  constant (limitation des divisions):
  - $Z_{ref} = d*x_{ref}+e*y_{ref}+f$
  - $A[z_{ref}] = a/z_{ref}$ ,  $b[z_{ref}] = b/z_{ref}$ ,  $c[z_{ref}] = c/z_{ref}$
  - $U[x+1,y-d/e] = U[x,y] + a[z_{ref}] - b*d/e...$

# Calcul incrémental

- Exercice : connaissant les trois sommets  $(X_0, Y_0)$   $(X_1, Y_1)$  et  $(X_2, Y_2)$  d'un triangle

Définir une méthode incrémentale permettant de déterminer si un point de l'écran appartient au triangle. (Tous les points de l'écran sont parcourus séquentiellement)

# Plan de la présentation (2)

- **Gérer la précision des calculs**
  - Virgule fixe/Virgule flottante
  - Calcul incrémental
  - **Calcul incrémental et rendu par tuile**
  - Rendu vidéo : cahier des charges
- **Gérer les opérations coûteuses en ressources matérielles**
  - Comment éliminer des divisions...

# Calcul incrémental et rendu par tuile

- Deux niveaux hiérarchiques (image, tuile)
- Deux niveaux incrémentaux
- Complexité matérielle : est ce toujours payant ?
- Exercice : évaluation de complexité en tenant compte de la précision des opérateurs (image/texture 512x512, transformée affine,  $\frac{1}{4}$  de pixel de précision)

















# Plan de la présentation (2)

- **Gérer la précision des calculs**
  - Virgule fixe/Virgule flottante
  - Calcul incrémental
  - Calcul incrémental et rendu par tuile
  - **Rendu vidéo : cahier des charges**
- **Gérer les opérations coûteuses en ressources matérielles**
  - Comment éliminer des divisions...



















# Plan de la présentation (2)

- Gérer la précision des calculs
  - Virgule fixe/Virgule flottante
  - Calcul incrémental
  - Calcul incrémental et rendu par tuile
  - Rendu vidéo : cahier des charges
- Gérer les opérations coûteuses en ressources matérielles
  - Comment éliminer des divisions...

















