M2-Images

Primitives non planes et Visibilité

J.C. Iehl

October 6, 2010

Objectif : REYES

ou une approximation ...

- dessiner des primitives non planes,
- les découper tant qu'elles ne sont pas complètement visibles,
- les transformer en primitives planes lorsque la transformation n'introduit pas d'erreur visible,
- + visibilité lorsque l'on dessine plusieurs primitives,
- + couleur des pixels ?

dans ce cours :

utilisation des PN Triangles, triangles de Bézier, primitives non planes les "plus simples".

PN Triangles : qu'est ce que c'est ?

qu'est ce que c'est ?

- une surface construite à partir de 3 sommets et des 3 normales associées,
- la "plus simple" : pas de données supplémentaires (triangles voisins, etc.)

les détails :

"Curved PN Triangles"
A. Vlachos, J. Peters, C. Boyd, J.L. Mitchell, 2001

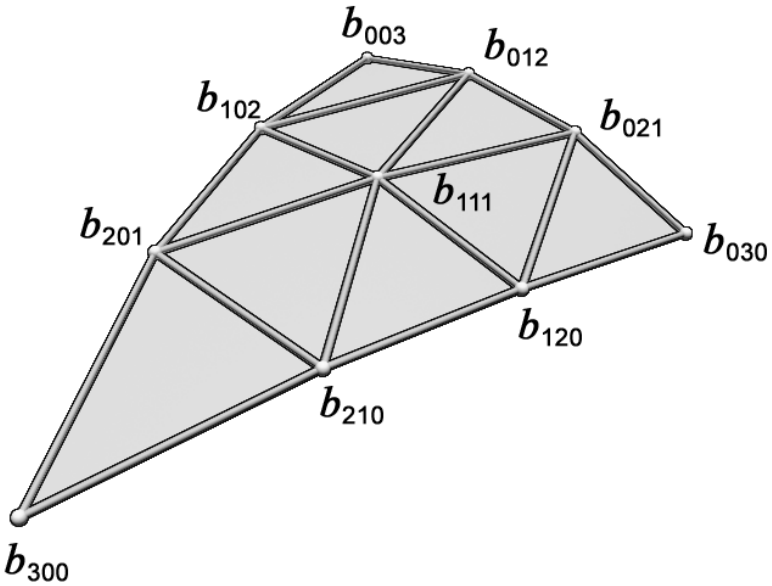
PN Triangles : comment ça marche ?

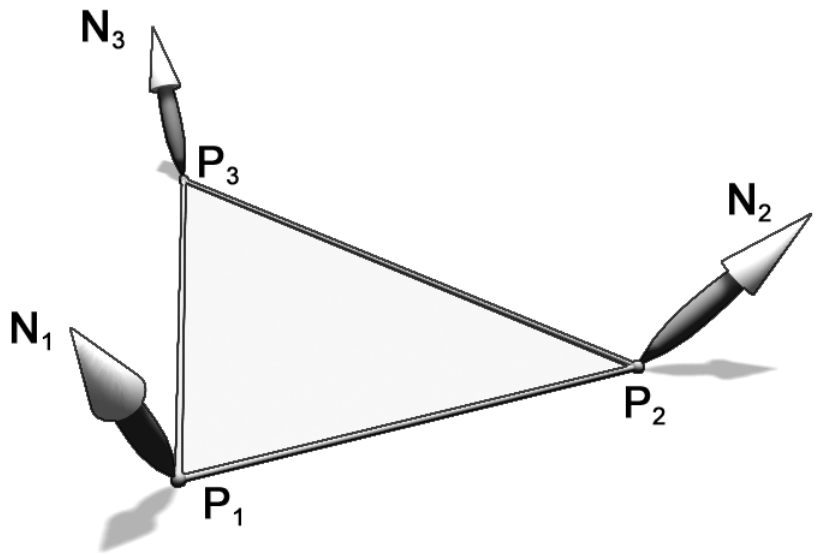
comment ça marche ?

- cf. le cours de modélisation pour les "vraies" explications sur les surfaces de Bézier,
- construction de 9 points de contrôle à partir des 3 sommets et de leurs normales,
- la surface est définie par un polynome de degré 3, utilisant les 9 points de contrôle,
- l'évaluation du polynome fournit un point sur la surface.

en résumé : "tirer" la surface vers les plans tangents.

comment ça marche ?





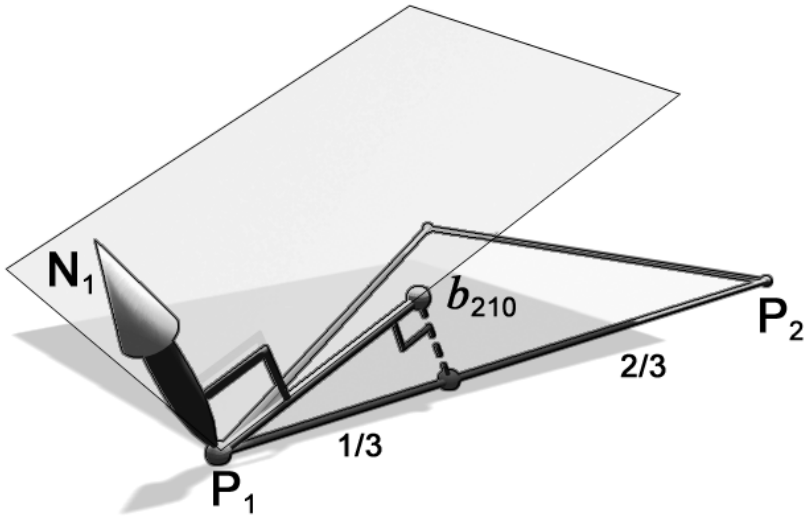
évaluation du polynome :

$$
\begin{aligned}
p(u,v)_{w\equiv 1-u-v} &= b_{300}w^3 + b_{030}u^3 + b_{003}v^3 \\
&+ b_{210}w^2u + b_{120}wu^2 + b_{201}w^2v \\
&+ b_{021}3u^2v + b_{102}3wv^2 + b_{012}3uv^2 \\
&+ b_{111}6wuv
\end{aligned}
$$

comment ça marche ?

construction des points de contrôle :

- à partir d'un sommet, de sa normale, et d'une arête,
- projette un point de l'arête sur le plan tangent,



comment ça marche ?

évaluer les points de la surface :

- la surface est définie sur un domaine paramétrique : (u, v, w),
- avec quelques contraintes :
- $u + v + w = 1$,
- $u \in [0\ 1]$, $v \in [0\ 1]$ et $w = 1 - u - v$,
- un domaine 2d : un triangle unitaire.

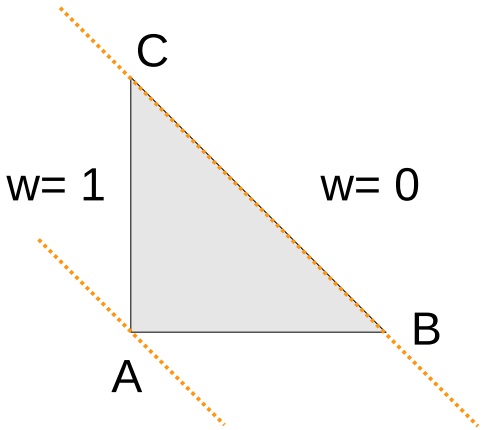
rappel :

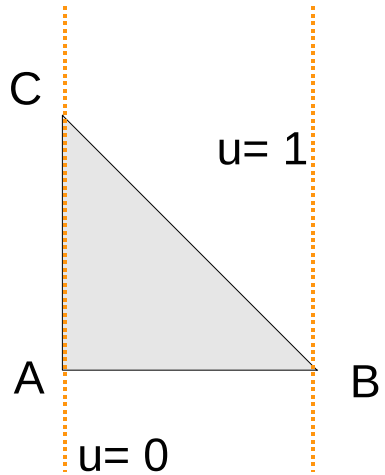
on peut évaluer les points de la surface d'un triangle abc de la même manière : $p(u, v)_{w \equiv 1-u-v} = wa + ub + vc$.

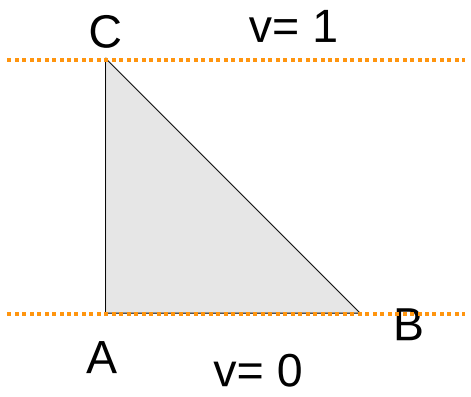
Evaluation : domaine paramétrique

domaine paramétrique :

- les sommets du triangle ont des positions dans le repère local,
- mais aussi dans le domaine paramétrique.
- $p(u = 0, v = 0, w \equiv 1) = a$,
- $p(u = 1, v = 0, w \equiv 0) = b$,
- $p(u = 0, v = 1, w \equiv 0) = c$.





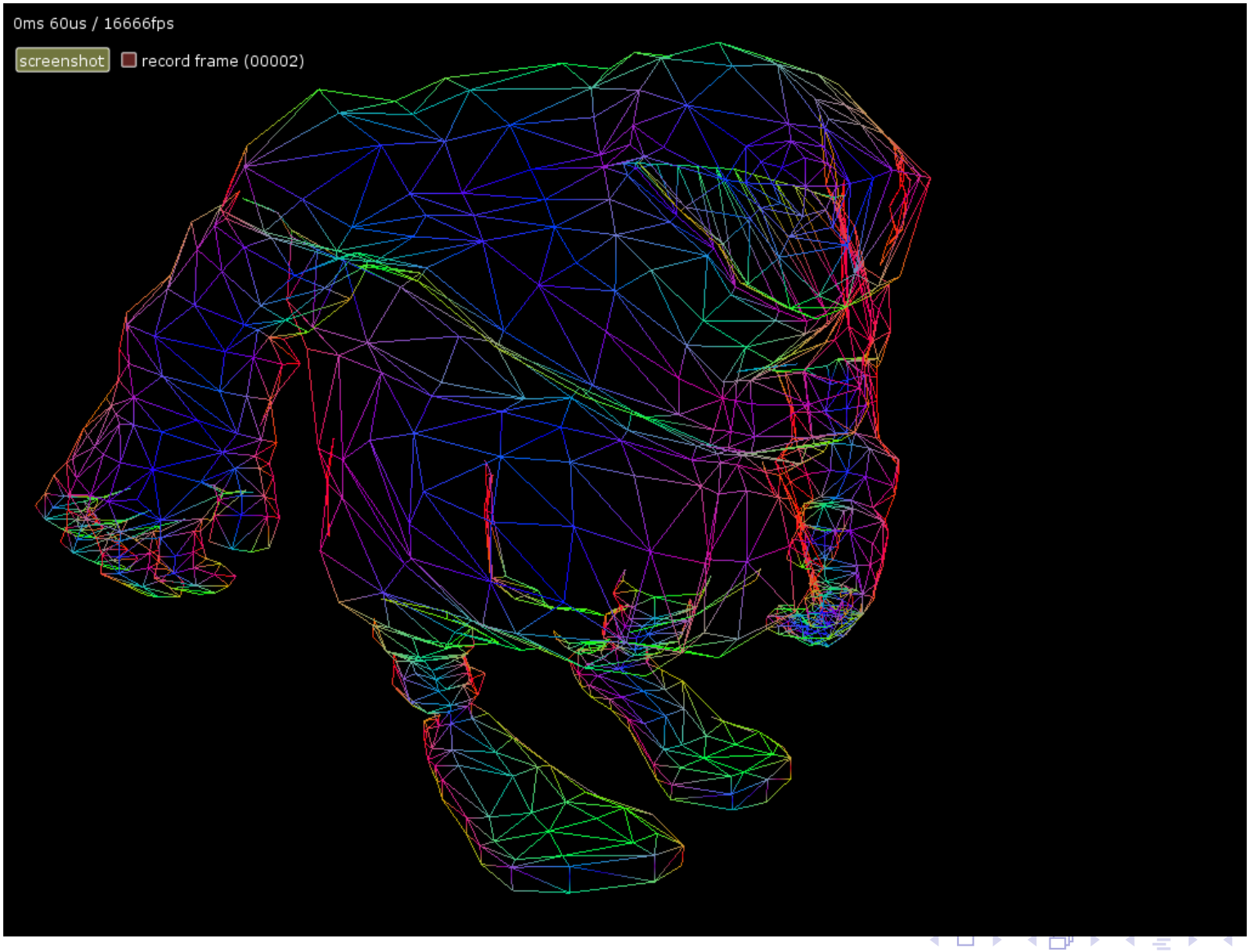


Evaluation : domaine paramétrique

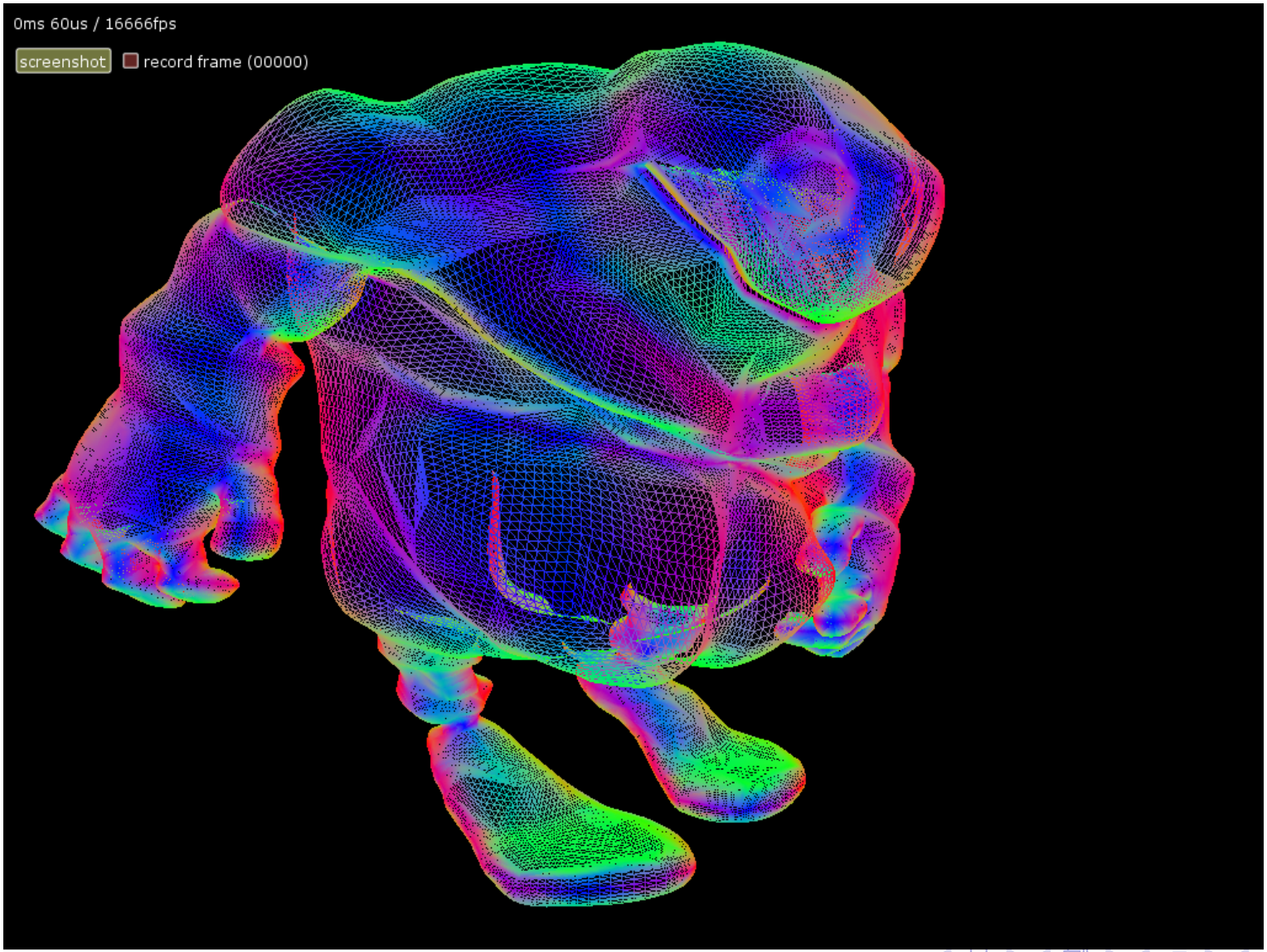
questions :

- quels paramètres (u, v) correspondent au milieu du triangle ?
- quelle position (x, y, z) correspond au milieu du triangle ?
- mêmes questions pour le milieu de chaque arete du triangle.
- quelle position (x, y, z) appartenant à la surface correspond au milieu du triangle ?
- mêmes questions pour le milieu de chaque bord de la surface.

exemple : triangles



exemple : PN Triangles, subdivision régulière



Subdivision : qu'est ce que c'est ?

pourquoi subdiviser ?

- pour afficher quelquechose qui ressemble à la surface du pn triangle . . .
- découper la surface (non plane) en morceaux plans,
- tant qu'un critère n'est pas satisfait.

par exemple : distance entre le point milieu du triangle et le point milieu de la surface $< d$.

REYES

"The REYES rendering architecture"
R.L. Cook, L. Carpenter, E. Catmull, 1987

principes de conception :

- simplifier les différentes étapes du pipeline graphique,
- pas de limite sur le nombre d'objet, sur le nombre de primitives par objet,
- produire des images aussi visuellement riches que le réel.

REYES : algorithme

dessiner un ensemble d'objets :

- pour chaque objet,
- pour chaque primitive (lire la primitive),
- déterminer la boite englobante de la primitive,
- éliminer les primitives non visibles,
- si la boite englobante est "trop grosse" :
  découper la primitive, recommencer,
- fragmenter la primitive en *micro-polygones* pour la dessiner.

définir : trop "grosse" ?

REYES : algorithme

micro-polygones :

- les micro-polygones sont plus petits qu'un pixel,
- cf. *tant que la boite englobante est trop grosse : découper*,
- ils sont donc générés par un "petit" morceau de surface,
- ils forment un découpage régulier du "petit" morceau de la surface,
- dessiner avec un z-buffer pour obtenir une visibilité correcte.

définir : trop "grosse" / "petit" morceau de surface ?

REYES : algorithme

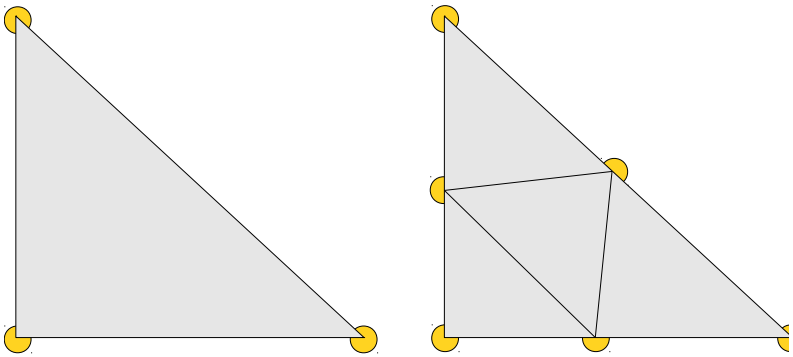
critere d'arret du découpage :

- principe : ne pas introduire d'erreur visible dans l'image,
- découper la primitive tant que les micro-polygones sont plus gros qu'un pixel.

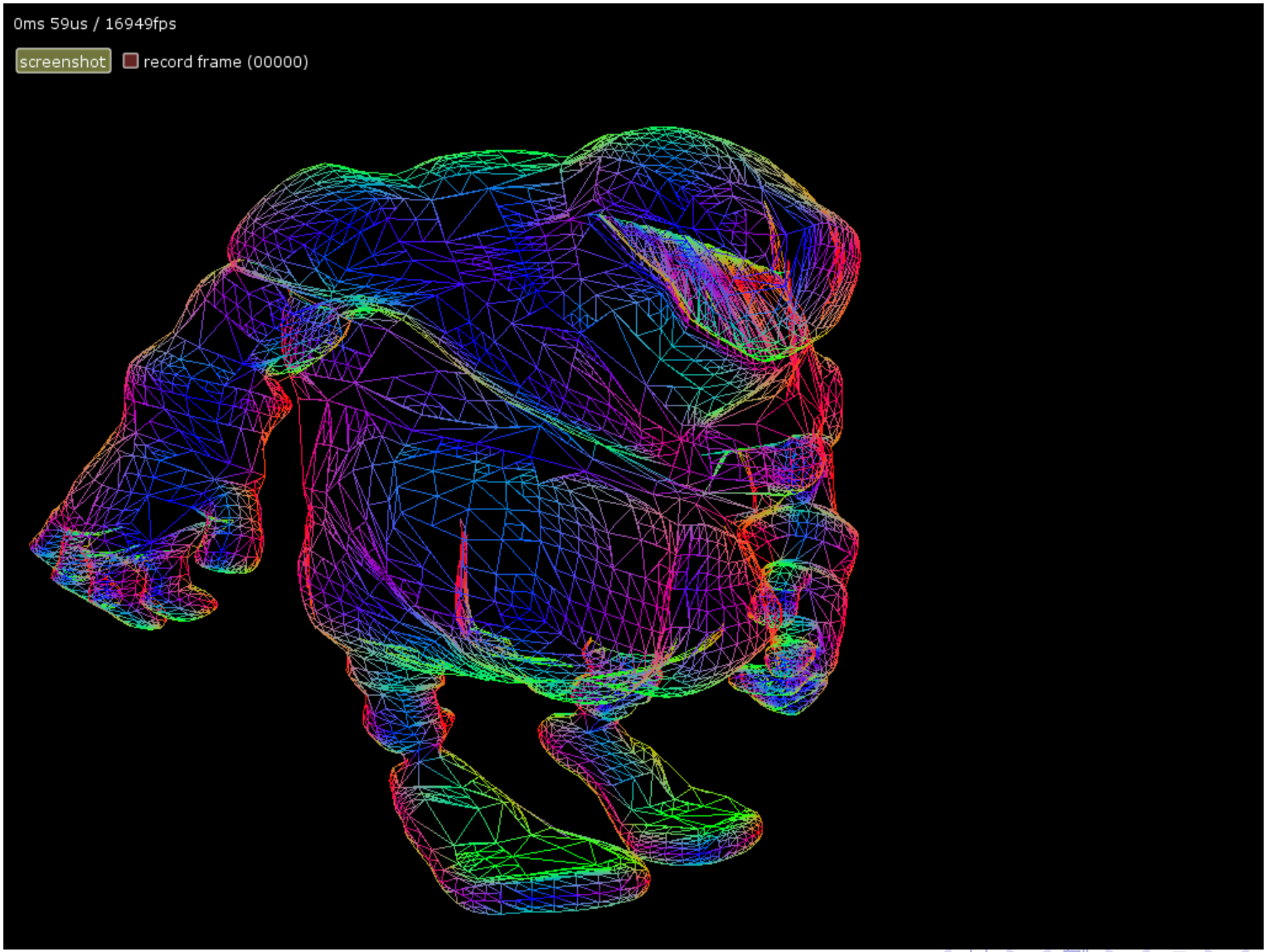
REYES : application aux PN Triangles

et alors ?

- pour chaque primitive (*PN Triangle*) :
- construire les points de contrôle,
- représenter les sommets de la surface dans le domaine paramétrique,
- découper la surface dans le domaine paramétrique, construire les 4 triangles,
- recommencer tant que . . .



exemple : PN Triangles, subdivision adaptative



REYES : application aux PN Triangles

recommencer tant que . . .

- un triangle visible se projette sur plus de pixels que le nombre de micro-polygones générés pour le dessiner,
- (un triangle est partiellement visible).

prévoir un critère d'arret :

- nombre de découpages du pn triangle initial,
- dimensions de l'englobant du triangle.

pour les cas ou le nombre de pixels n'est pas calculable (triangle partiellement visible).

REYES : application aux PN Triangles

mais :

il reste encore quelques détails à regler . . .

exemple : PN Triangles, subdivision adaptative