M2-Images

Intersections

J.C. Iehl

November 12, 2009

Objectif : lancer de rayons

algorithme :

- déterminer la direction de la droite passant par l'observateur et le centre du pixel,
- calculer l'intersection de la droite avec tous les objets de la scène,
- ne conserver que la plus petite : l'objet visible, le plus proche de l'observateur,
- donner une couleur au point.

Objectif: calcul d'intersections

cas simples :

- plan,
- triangle,
- sphere,
- cube.

Intersection : rayon / plan

rappels :

- $p(t) = o + t \cdot \vec{d}$,
- $\vec{n} \cdot \overrightarrow{ap} = 0$ sur le plan de normale $\vec{n}$ passant par a et $p \in plan(a, \vec{n})$.

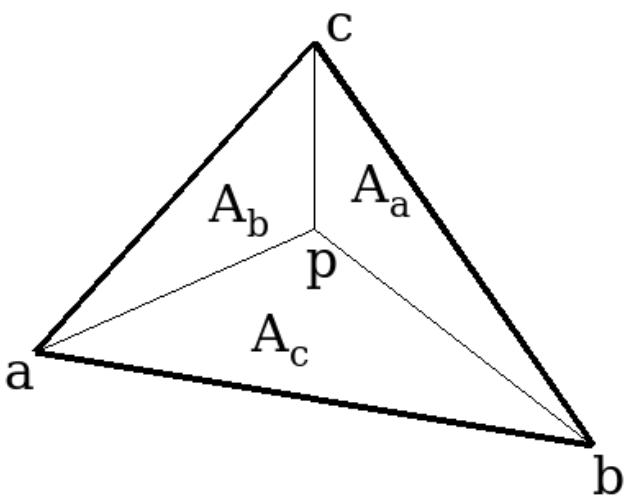
résultat :

- $\vec{n} \cdot \overrightarrow{ap(t)} = 0$
- $\vec{n} \cdot ((o + t \cdot \vec{d}) - a) = 0$
- $\vec{n} \cdot ((o - a) + t \cdot \vec{d})) = 0$
- $t = \frac{(a-o) \cdot \vec{n}}{\vec{d} \cdot \vec{n}}$

Intersection : rayon / triangle

rappels :

- $p(t) = o + t \cdot \vec{d}$,
- $p(\alpha, \beta) = \alpha a + \beta b + (1 - \alpha - \beta)c$ si $p \in triangle(a, b, c)$
- $\alpha = A_a/A$, $\beta = A_b/A$
- $\gamma = 1 - \alpha - \beta = A_c/A$



Intersection : rayon / triangle

résultat :
"Fast, Minimum Storage Ray-Triangle Intersection"
code + détails

Intersection : rayon / sphere

rappels :

- $p(t) = o + t \cdot \vec{d}$,
- $(p_x - c_x)^2 + (p_y - c_y)^2 + (p_z - c_z)^2 - R^2 = 0$ si $p \in sphere(c, R)$
- $(p - c) \cdot (p - c) - R^2 = 0$

résultat :

- $(o + t \cdot \vec{d} - c) \cdot (o + t \cdot \vec{d} - c) - R^2 = 0$
- $(\vec{d} \cdot \vec{d})t^2 + 2\vec{d} \cdot (o - c)t + (o - c) \cdot (o - c) - R^2 = 0$

Intersection : rayon / boite orientée

rappels :

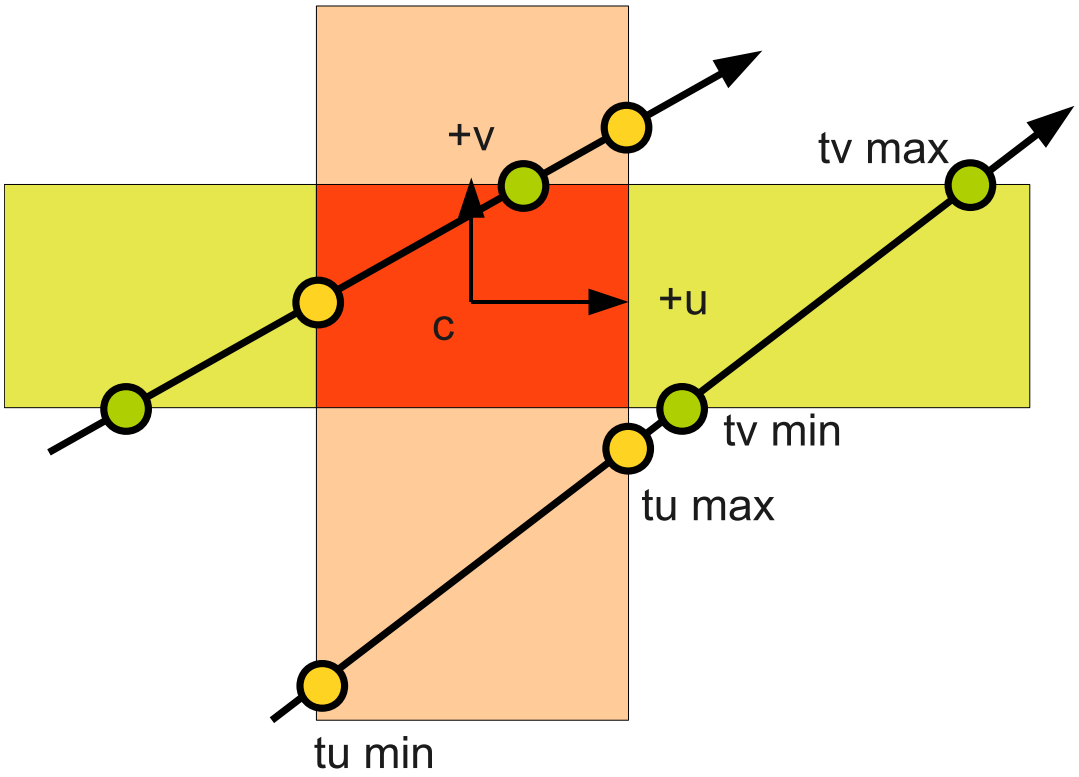
- $p(t) = o + t \cdot \vec{d}$,
- $p \in boite(c, \vec{u}, \vec{v}, \vec{w})$
- la boite est l'intersection de 3 paires de plans parallèles :
- $plan(c - \vec{u}, -\vec{u})$, $plan(c + \vec{u}, \vec{u})$,
- $plan(c - \vec{v}, -\vec{v})$, $plan(c + \vec{v}, \vec{v})$,
- $plan(c - \vec{w}, -\vec{w})$, $plan(c + \vec{w}, \vec{w})$,

résultat :

- calculer t_{min} et t_{max} pour chaque paire de plans (cf. intersection rayon / plan),

Intersection : rayon / boite orientée

- l'intersection existe si l'intersection des intervalles n'est pas vide : $max(t^u_{min}, t^v_{min}, t^w_{min}) < min(t^u_{max}, t^v_{max}, t^w_{max})$



Intersection : rayon / boite alignée sur les axes

détails et astuces de calculs :
"An Efficient and Robust Ray-Box Intersection Algorithm"
code

Objectif : plusieurs objets

pour chaque pixel (x, y)

- générer le rayon $r_{(x,y)} = (o(x, y), \overrightarrow{d(x, y)})$,
- pour chaque objet :
- calculer l'intersection du rayon et de l'objet, t_{objet}
- si $t_{objet} < t_{min}$
- $t_{min} = t_{objet}$
- calculer la position du point le long du rayon : $r_{(x,y)}(t_{min})$

Objectif : un peu de lumière

selon son orientation, une surface reçoit plus ou moins de lumière : $L_i(p) = L(\vec{l}) \cos\theta$ avec θ l'angle entre $\vec{l}$ et $\vec{n}_p$

selon sa matière et son orientation, une surface réfléchit plus ou moins de lumière vers l'observateur : notation $f_r(\vec{l}, p, \vec{o})$.

Objectif : un peu de lumière

BRDF :

- aspect mat / diffus : $f_r(\vec{l}, p, \vec{o}) = (0 < \mathit{constante} < 1)$,
- aspect réfléchissant: $f_r(\vec{l}, p, \vec{o}) = \frac{m+2}{2} \cos^m \theta_h$
  avec $\vec{h} = \frac{1}{2}(\vec{l} + \vec{o})$ et θ_h l'angle entre $\vec{n}_p$ et $\vec{h}$,

résultat :

- $L_r(p, \vec{o}) = L_i(\vec{l}) f_r(\vec{l}, p, \vec{o})$,
- $L_r(p, \vec{o}) = L(\vec{l}) \cos\theta \; f_r(\vec{l}, p, \vec{o})$.

Objectif : un peu d'ombre

comment déterminer qu'un point est éclairé par une source de lumière ?

idée :
vérifier la visibilité du point et d'un point de la source.

générer un autre rayon et vérifier la présence d'un objet plus proche que la source.

- s'il y a une intersection : un autre objet masque la source, le point est donc à l'ombre $L(\vec{l}) = 0$,
- sinon, le point est éclairé, il reçoit la lumière émise par la source.

Objectif : un peu de reflet

la lumière se réfléchit sur un miroir uniquement dans la direction $\vec{r} = 2(\vec{n} \cdot \vec{l})\vec{n} - \vec{l}$.

calculer l'énergie réfléchie par le miroir vers l'observateur ?

idée :
elle dépend de l'objet visible dans la direction $\vec{r}$.

générer un rayon dans la direction $\vec{r}$ et déterminer l'énergie réfléchie par le point visible.

Objectif : un peu de reflet

et s'il y a plusieurs miroirs ?

on recommence ...

l'algorithme devient récursif :

- pour connaître $L(\vec{r})$ l'énergie arrivant depuis le point visible dans la direction $\vec{r}$,
- il faut connaître l'énergie arrivant sur ce point, qui peut lui même être sur un miroir ... qui peut ...

Objectif : un peu de transparence

et si l'objet est transparent ?

- l'énergie réfléchie par un point à la surface d'un objet transparent dépend de l'énergie dans la direction $\vec{r}$,
- et de l'énergie dans la direction $\vec{t}$