Promenade en Informatique Graphique

en compagnie des mathématiques

Frédéric Mora

enseignant-chercheur en informatique (graphique) à l'université de Limoges
Membre du laboratoire XLIM (UMR CNRS 7252)
Responsable du département Métiers du Multimédia et de l'Internet à l'IUT du Limousin

Promenade en Informatique Graphique

en compagnie des mathématiques

Ce qu’on va (essayer de) faire :

- découvrir le domaine via plusieurs aspects
- montrer la diversité des maths utilisés

Ce qu’on ne fera pas :

- un cours de math
- un cours d’informatique

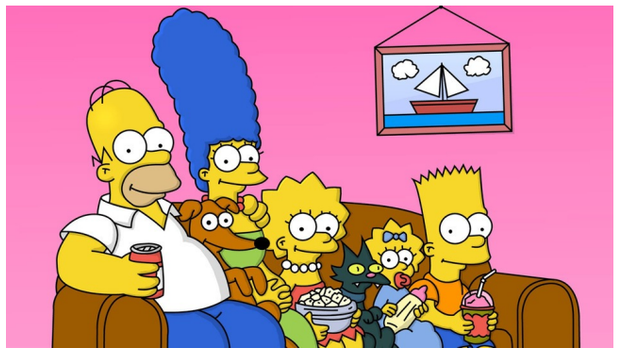
Informatique Graphique > c’est à dire ?

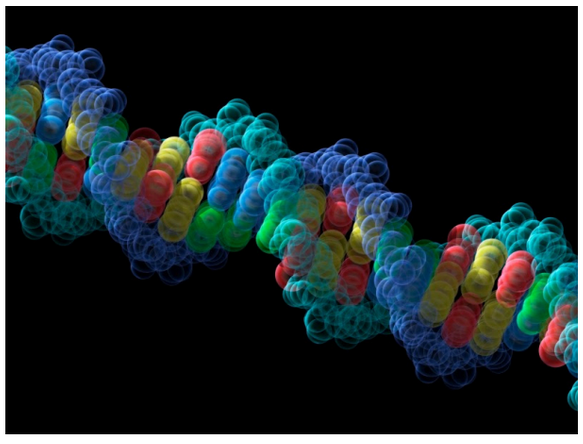
Qu’est-ce ?

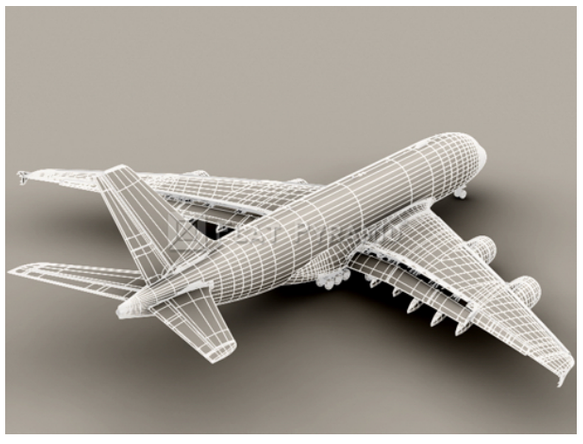
- un domaine de l’informatique

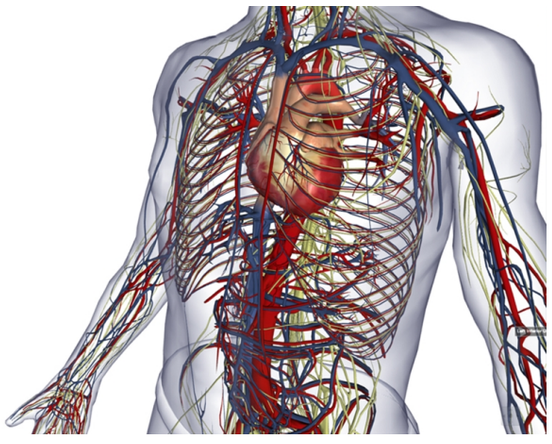
Ca sert à quoi ?

- créer des images et des films par ordinateur









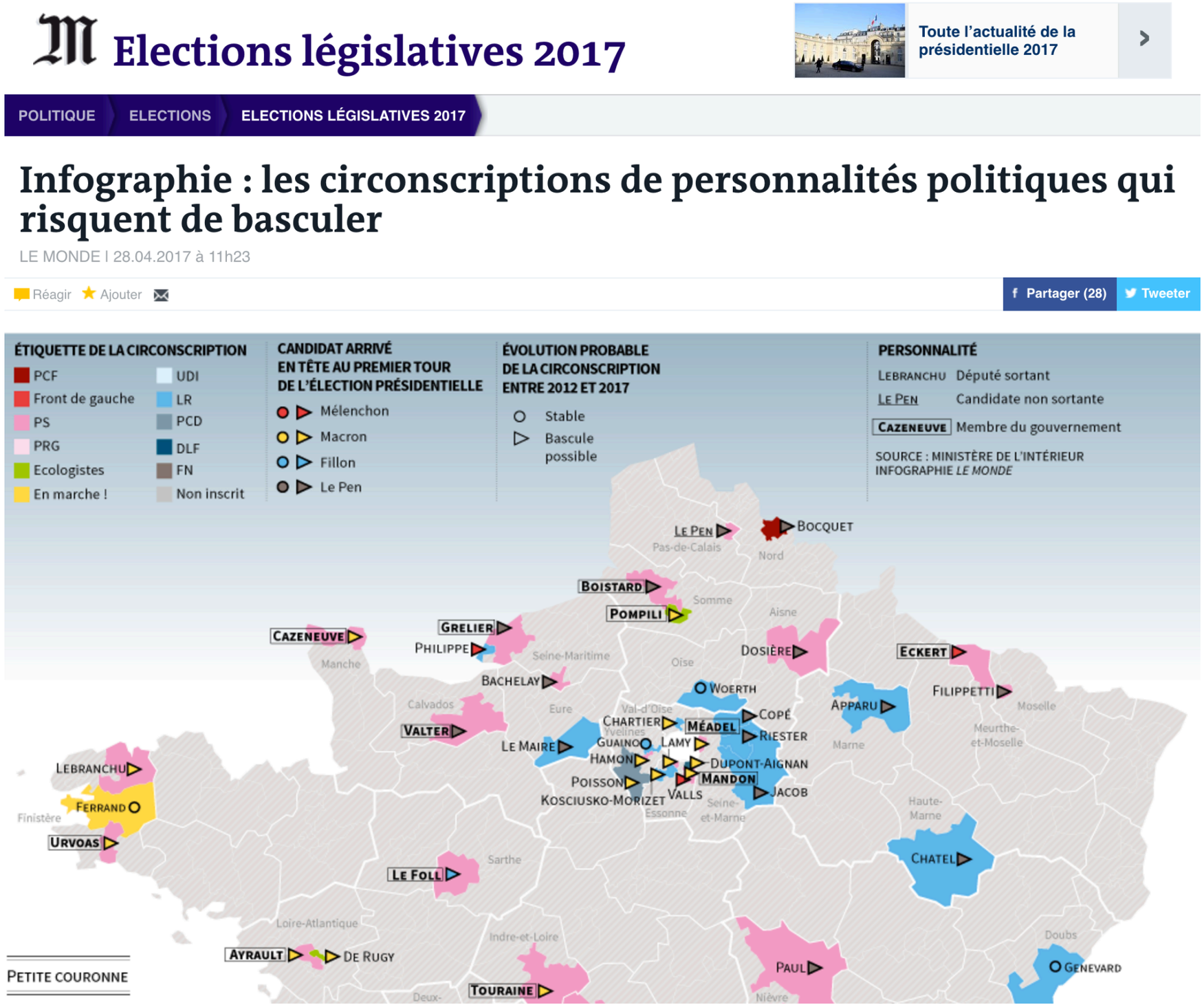
Informatique Graphique > c’est à dire ?

Infographie

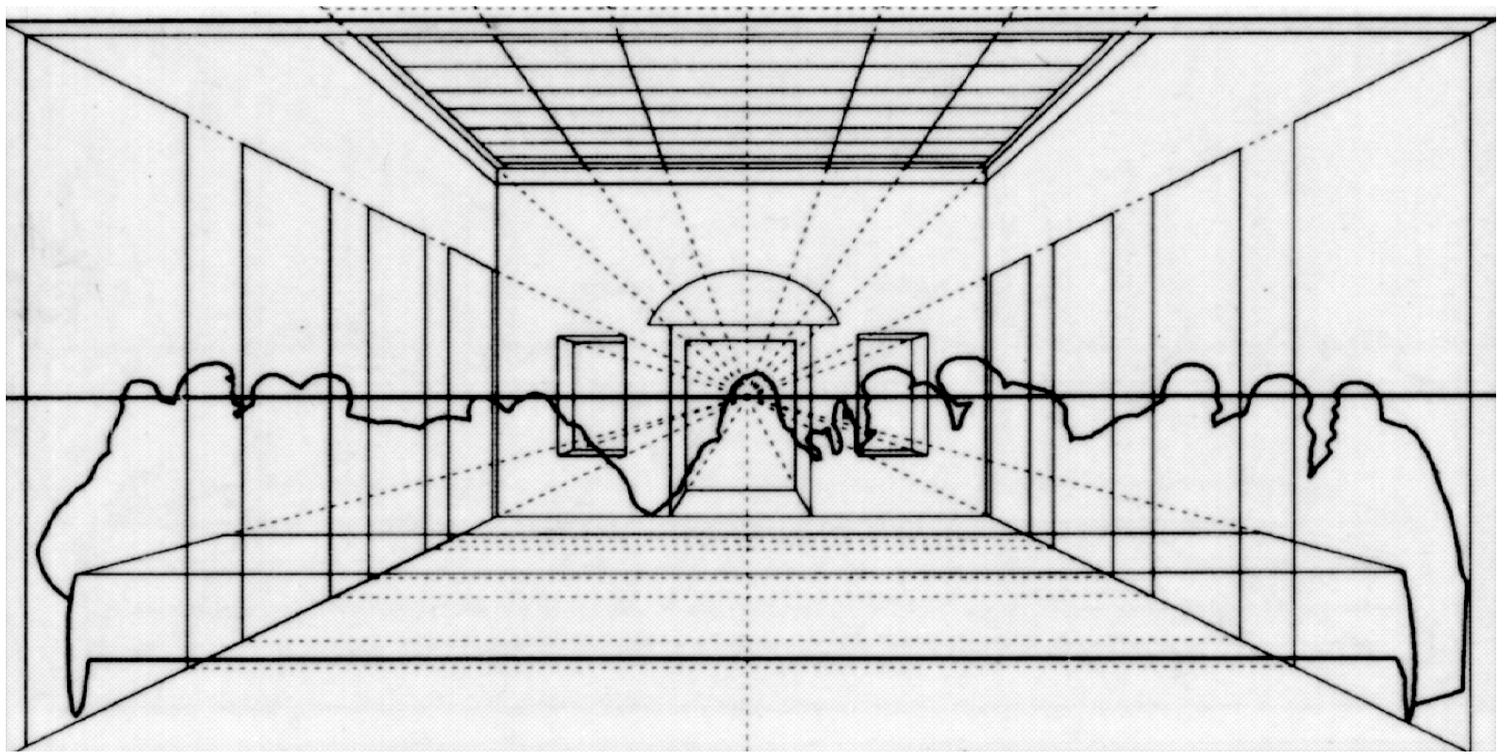
Infographics

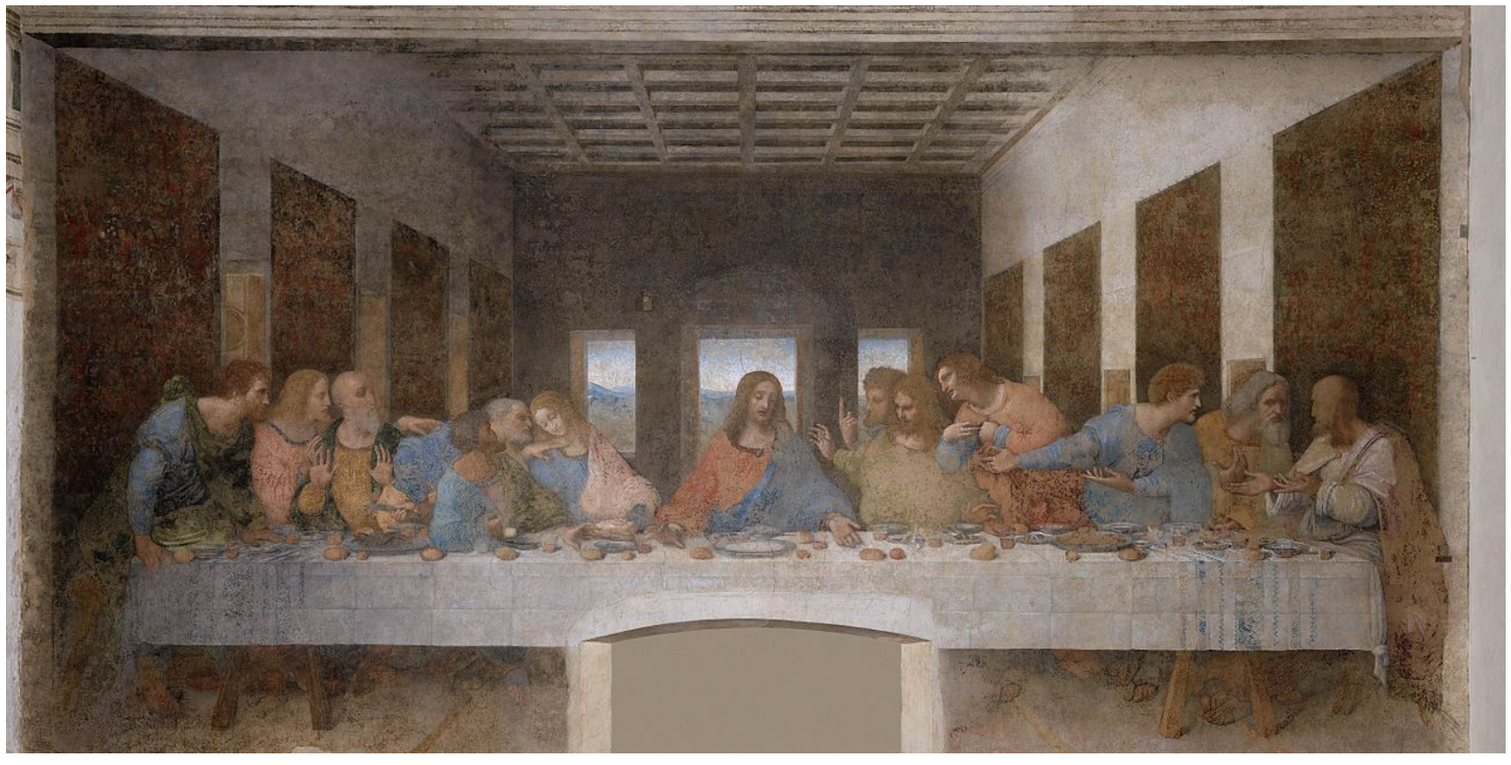
Information graphics

Ambigu

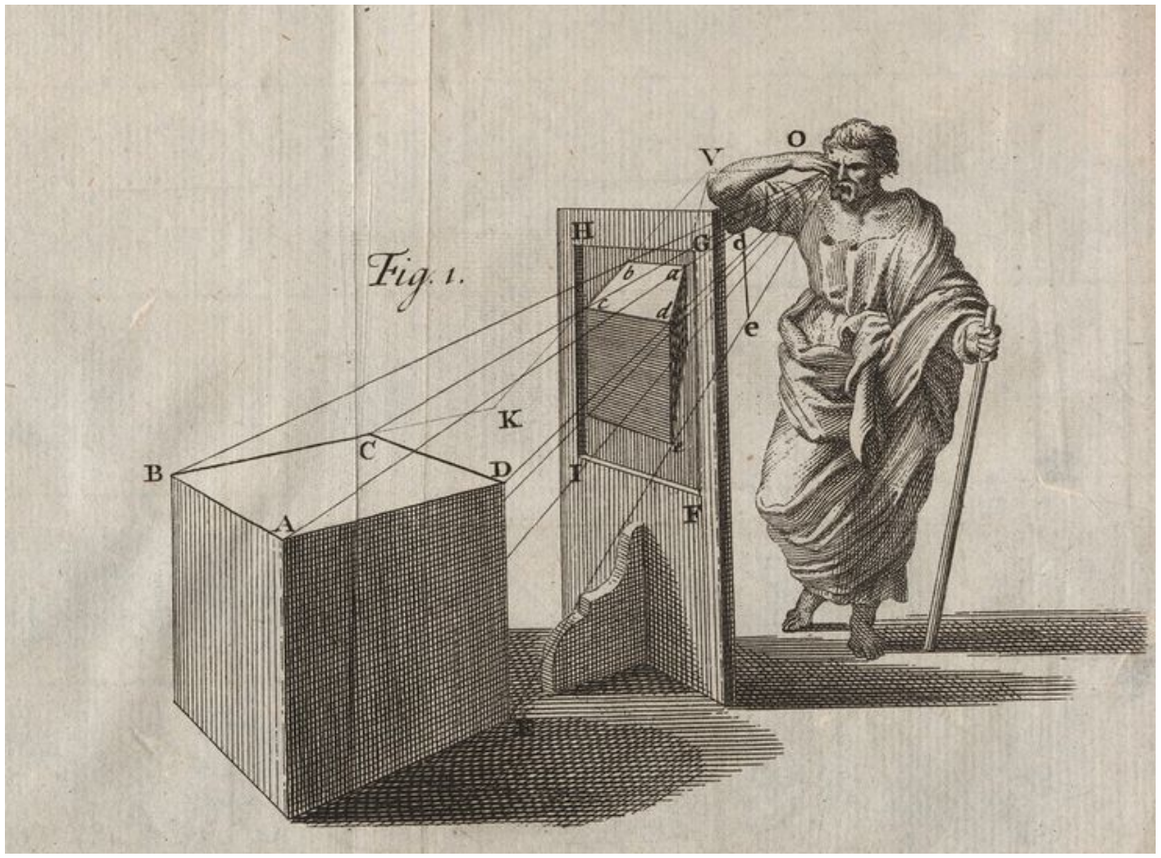


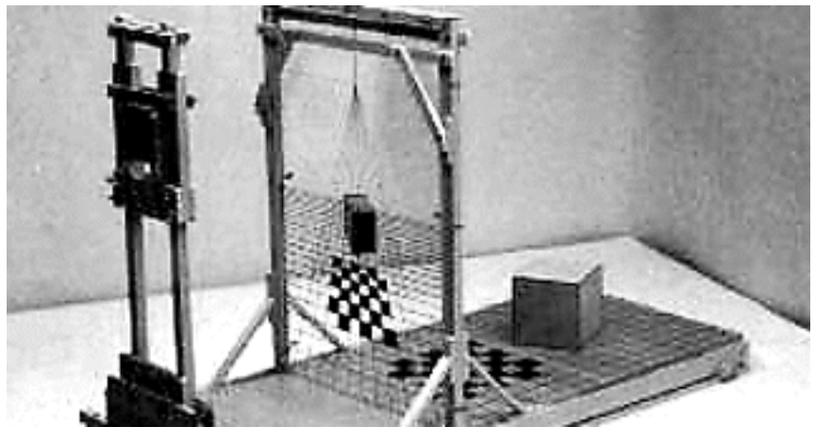
Informatique Graphique > (très très) brève histoire de la 3D





Informatique Graphique > (très très) brève histoire de la 3D





La première « machine » à faire de la 3D

- le perspectographe
- l’essentiel est déjà là

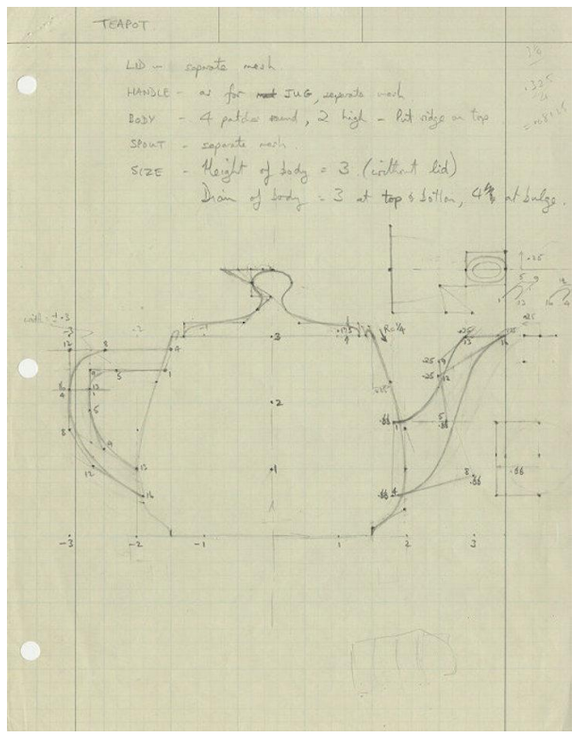
Informatique Graphique > (très très) brève histoire de la 3D

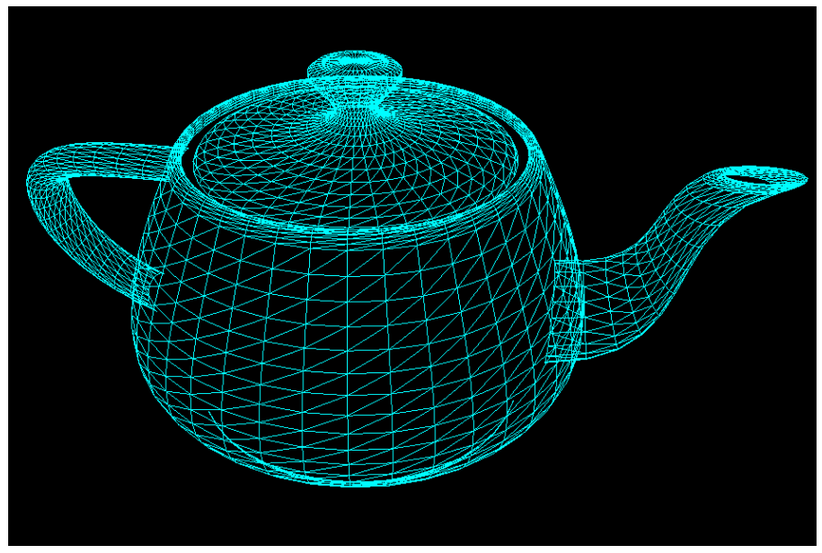
Naissance de la 3D moderne

- dans les années 1950, au MIT pour l'armée US (merci la guerre froide…)


whirlwind

- 1967, Université de l'Utah, modélisation d'objet 3D
- 1975, the Utah/Newell Teapot



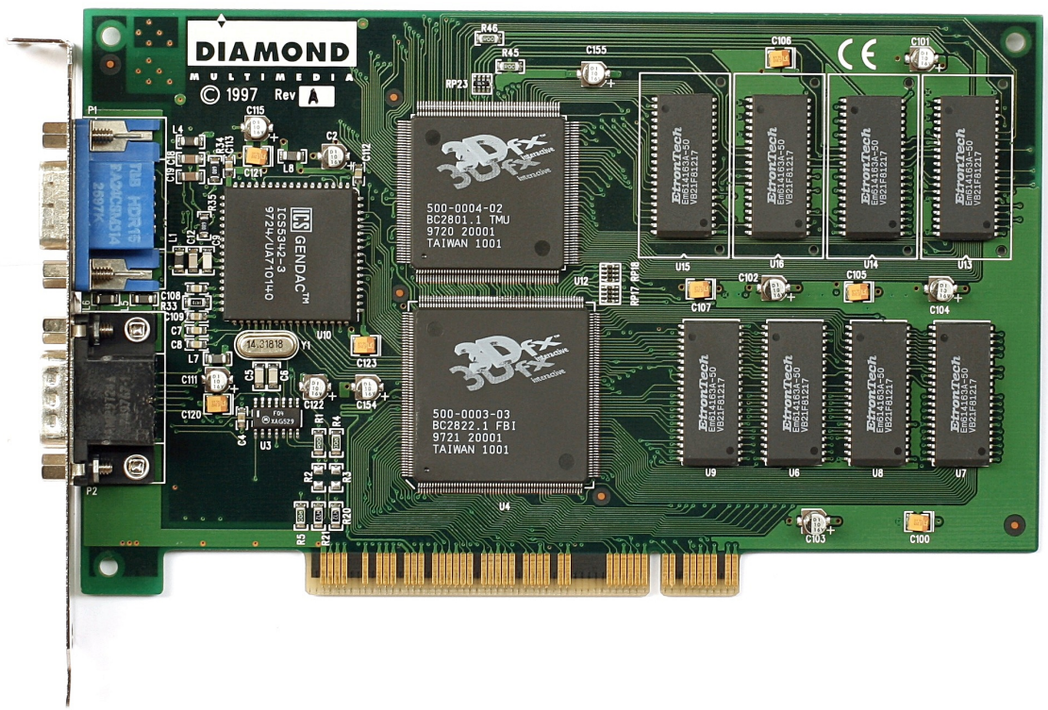


Informatique Graphique > (très très) brève histoire de la 3D

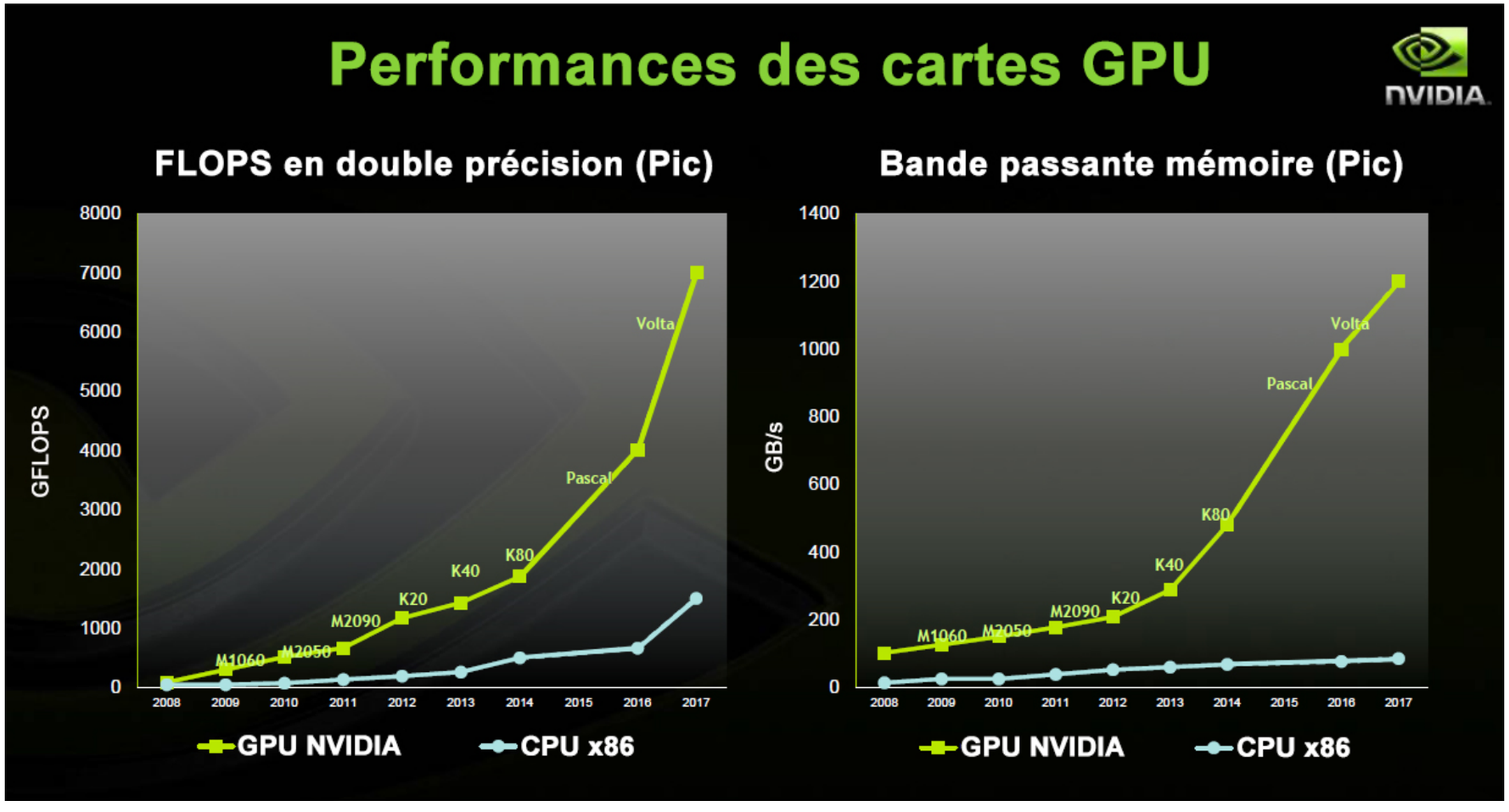
- la 3D coûte terriblement cher, réservée à un usage industriel ou militaire)

- année 1980, chute des coût (IBM-PC, Macintosh)

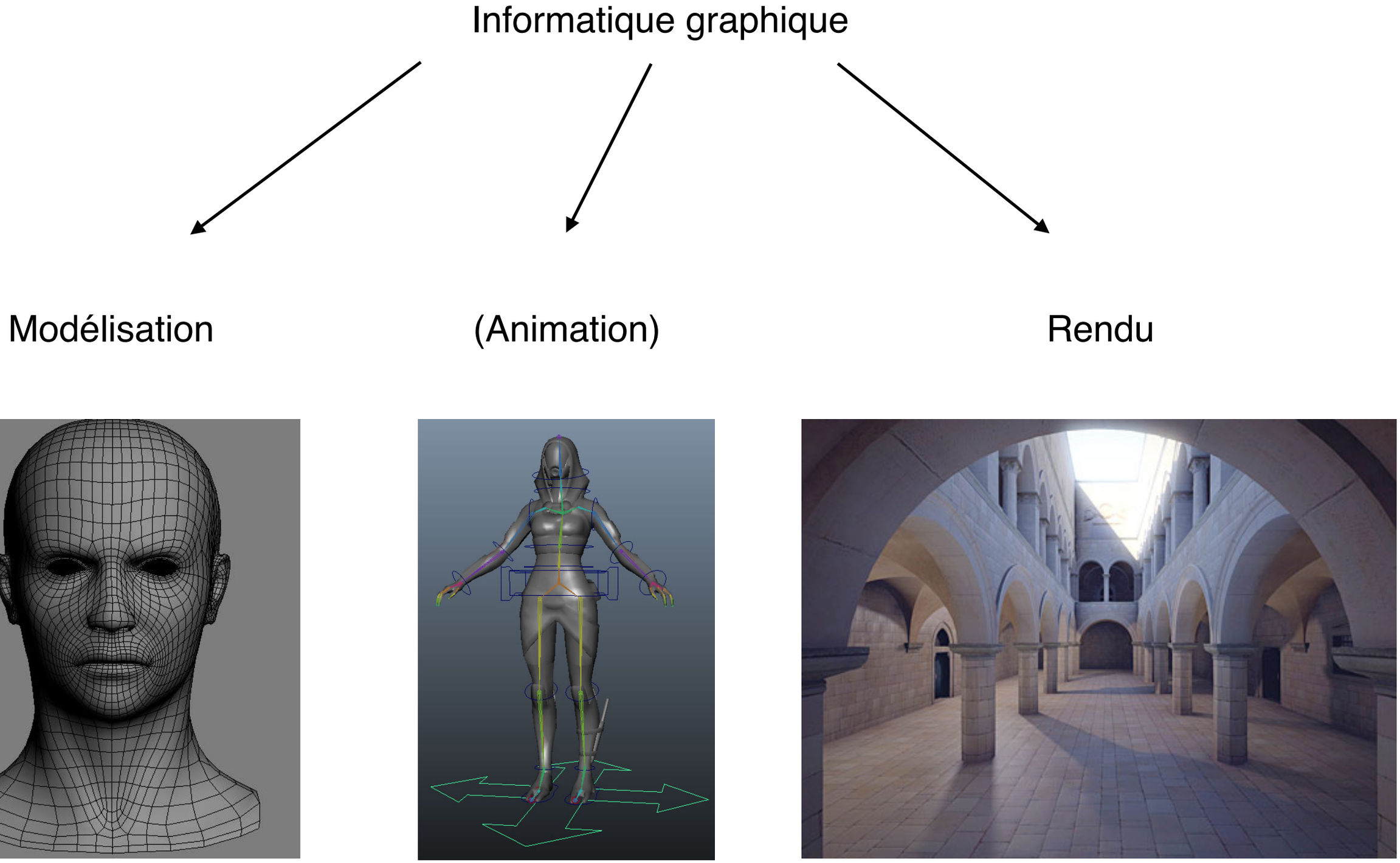
- Année 1990, apparition des cartes accélératrices 3D grand public (Voodoo 3DFX, Playstation, Dreamcast…)



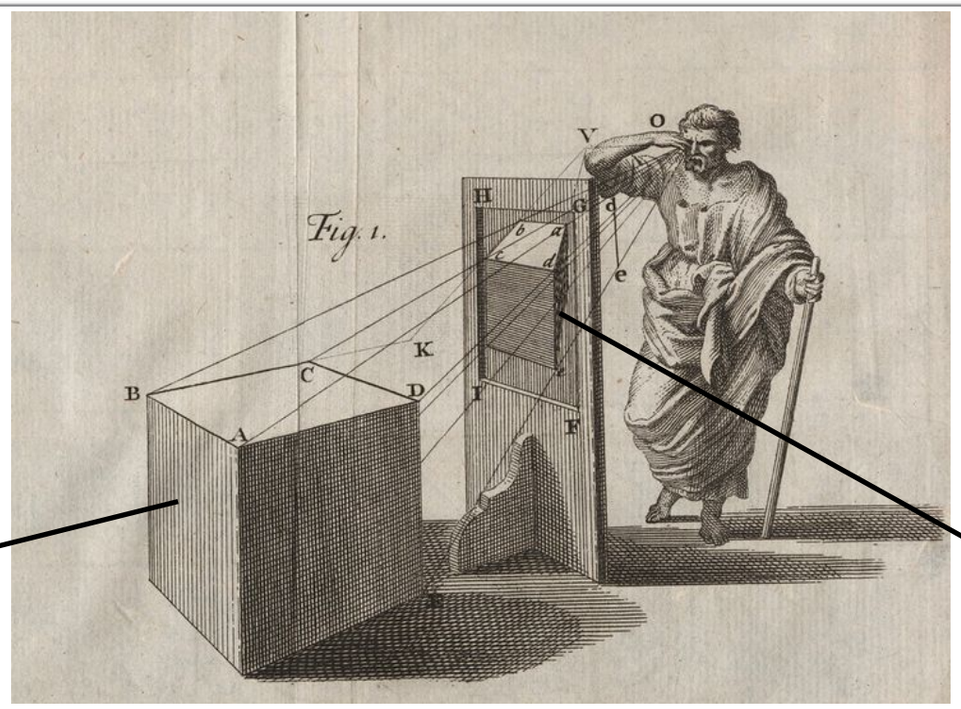
Informatique Graphique > (très très) brève histoire de la 3D



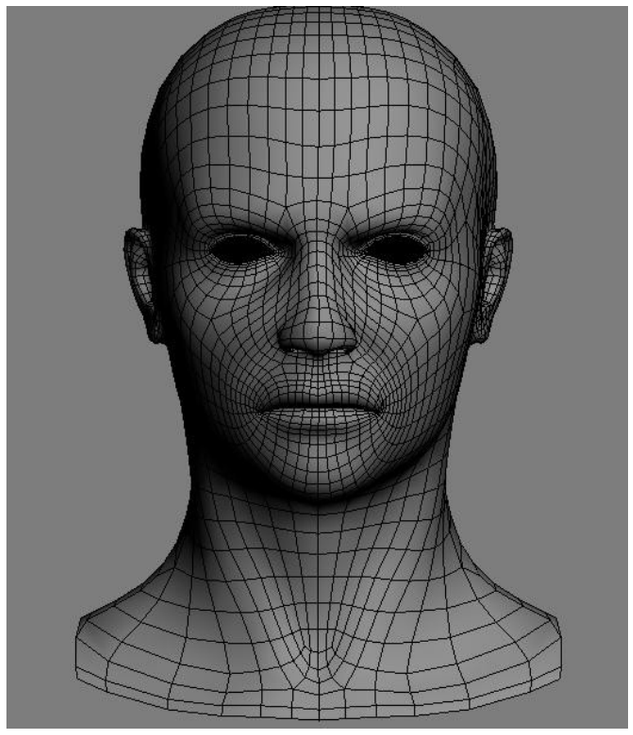
Informatique Graphique > domaine de recherche



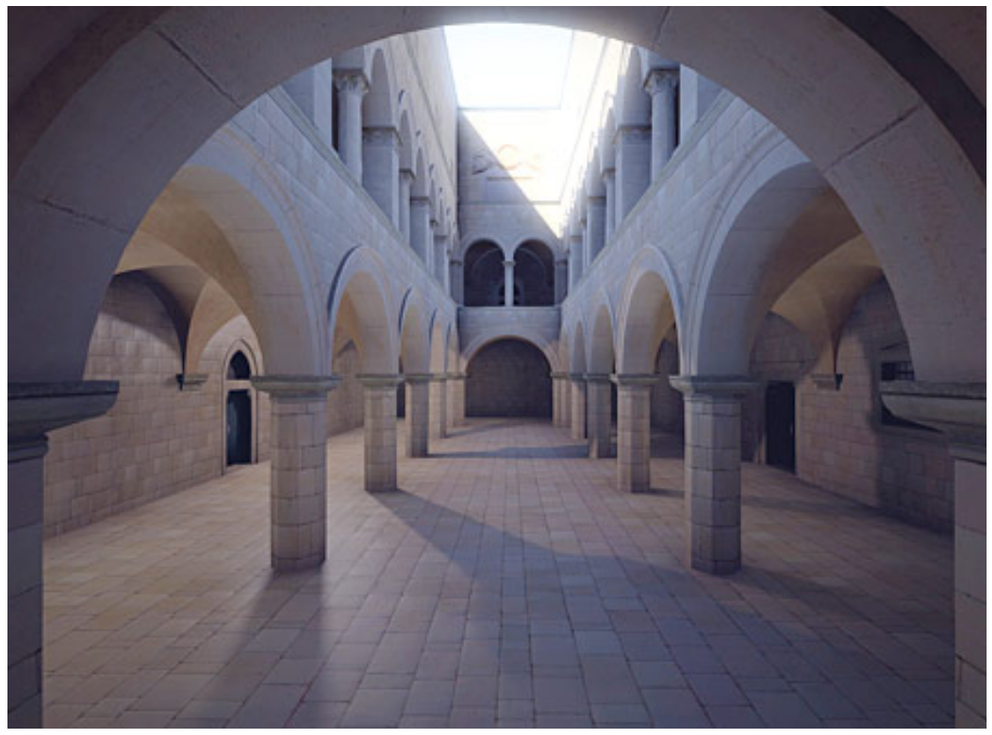
Informatique Graphique > domaine de recherche



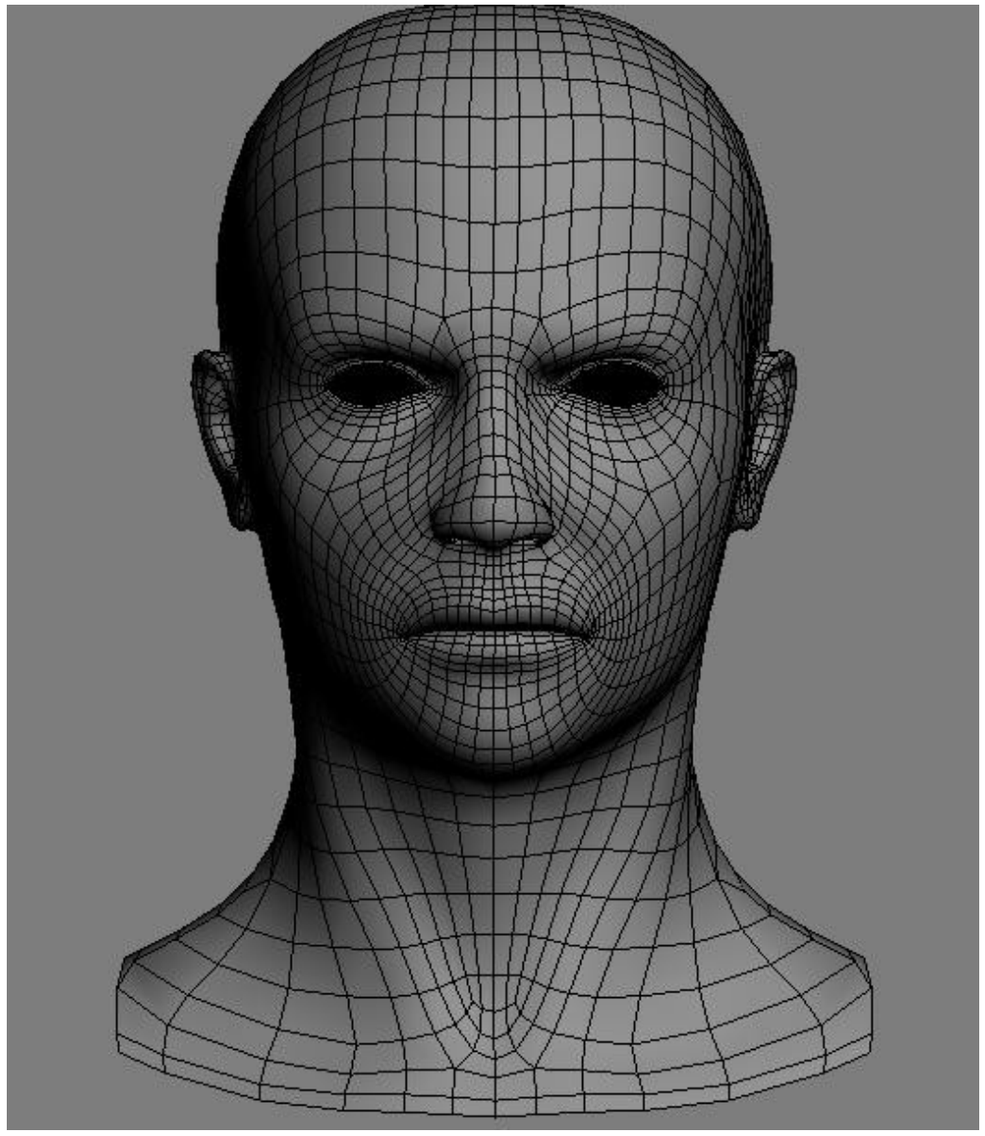
Modélisation



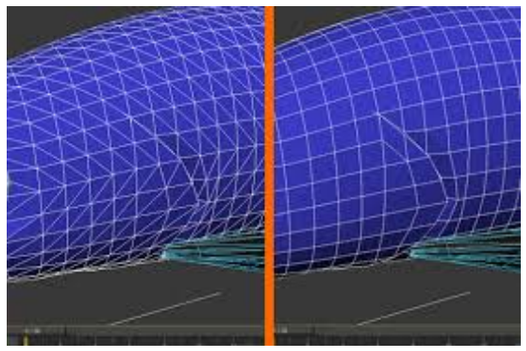
Rendu

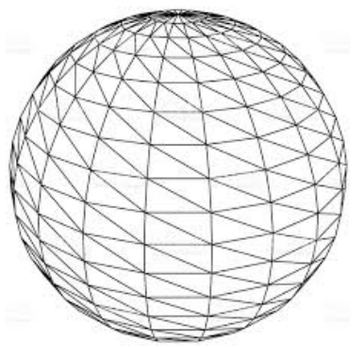


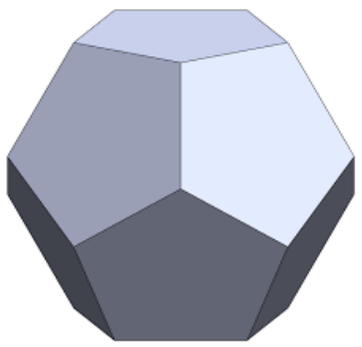
Informatique Graphique > modélisation

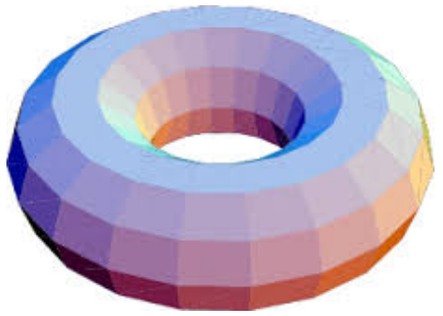


Modélisation géométrique > quads, triangles et primitives



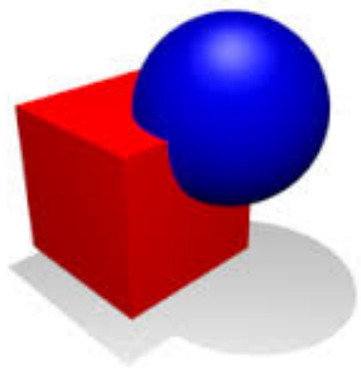




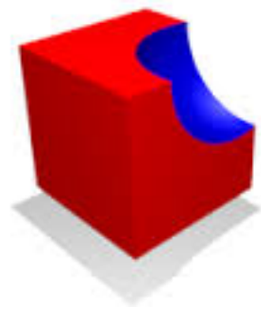


Géométrie dans l’espace

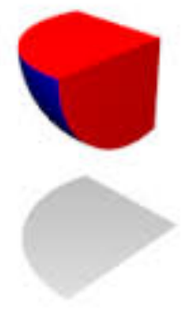
Modélisation géométrique > opérations CSG (algèbre de Boole)



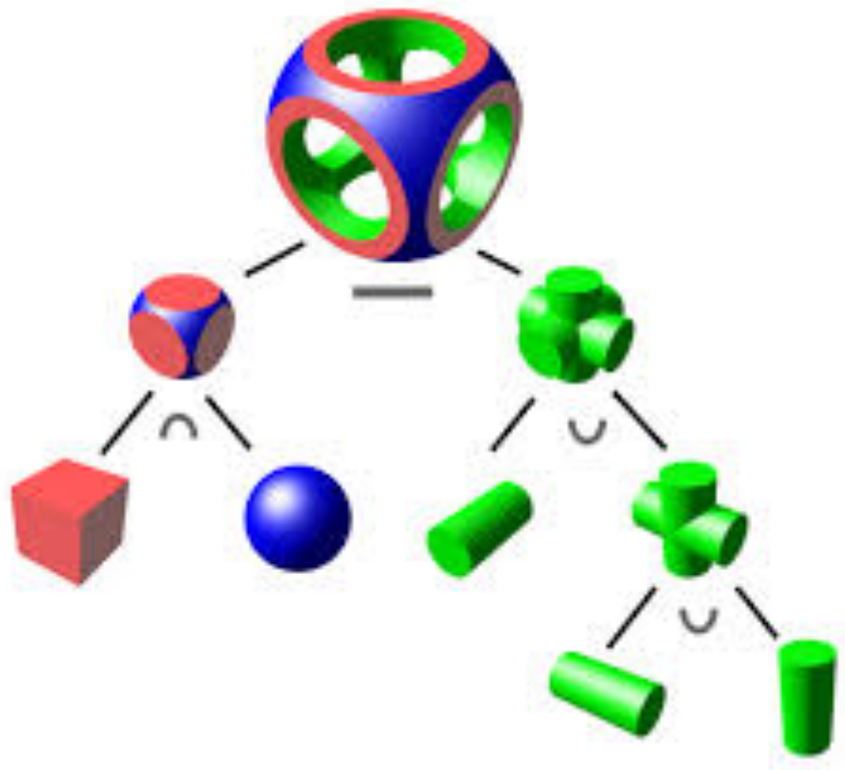
union



soustraction

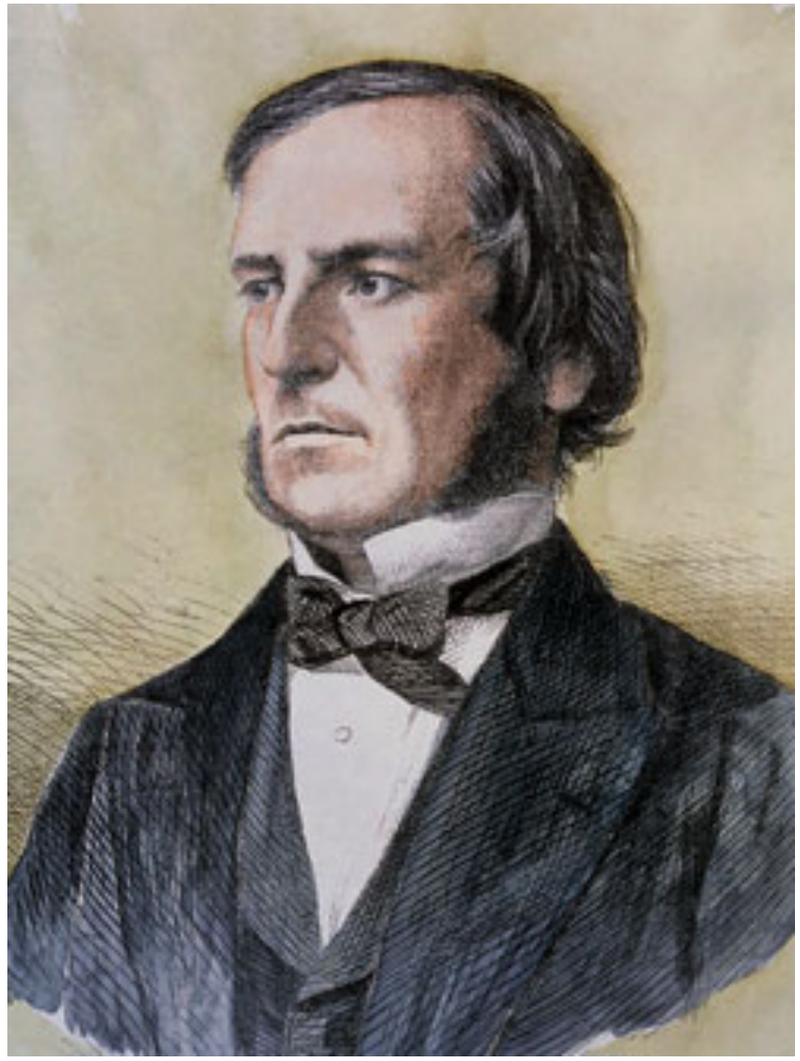


intersection



Modélisation géométrique > George Boole

George Boole, né le 2 novembre 1815 à Lincoln (Royaume-Uni) et mort le 8 décembre 1864 à Ballintemple (Irlande), est un logicien, mathématicien et philosophe britannique. Il est le créateur de la logique moderne, fondée sur une structure algébrique et sémantique, que l'on appelle algèbre de Boole en son honneur.



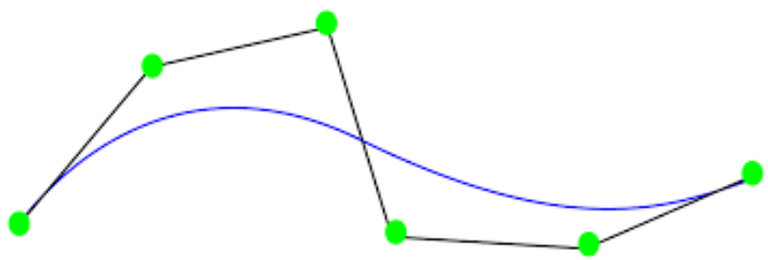
De 1844 à 1854, il crée une algèbre binaire, dite booléenne, n'acceptant que deux valeurs numériques : 0 et 1. Cette algèbre aura de nombreuses applications en téléphonie et en informatique, notamment grâce à Claude Shannon en 1938, près d'un siècle plus tard.

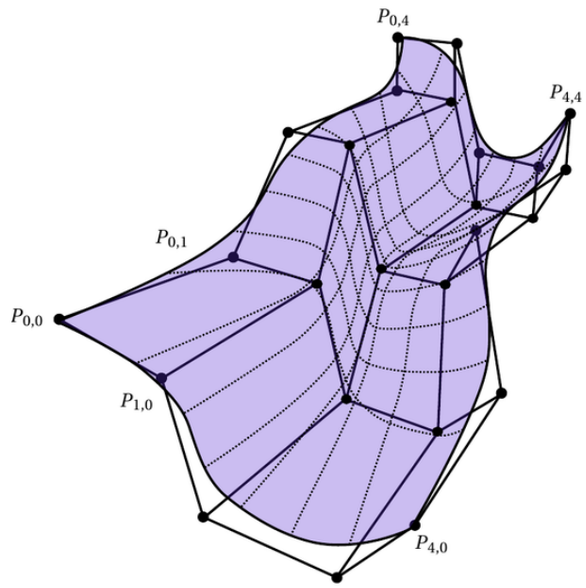
WIKIPEDIA
The Free Encyclopedia

Modélisation géométrique > Non Uniform Rational B-Splines

Spline
Bézier
→ B-Spline → NURBS

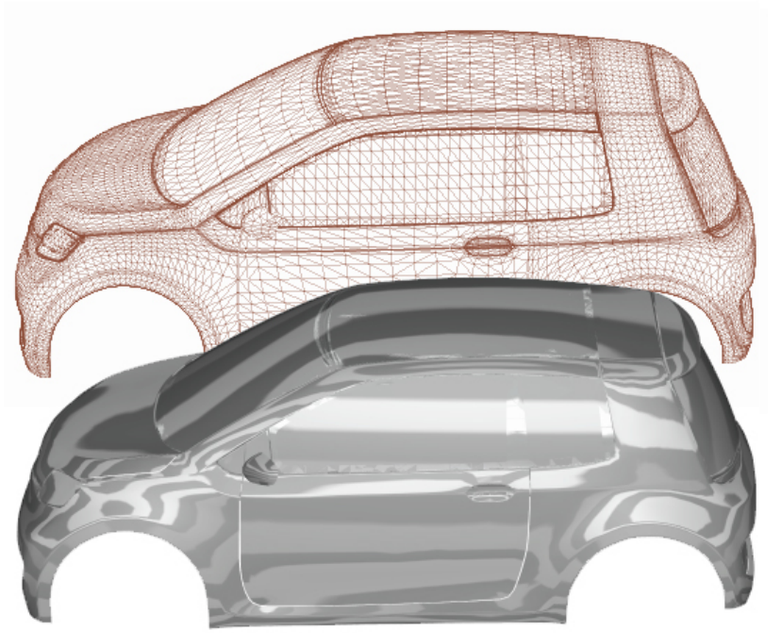
Courbe



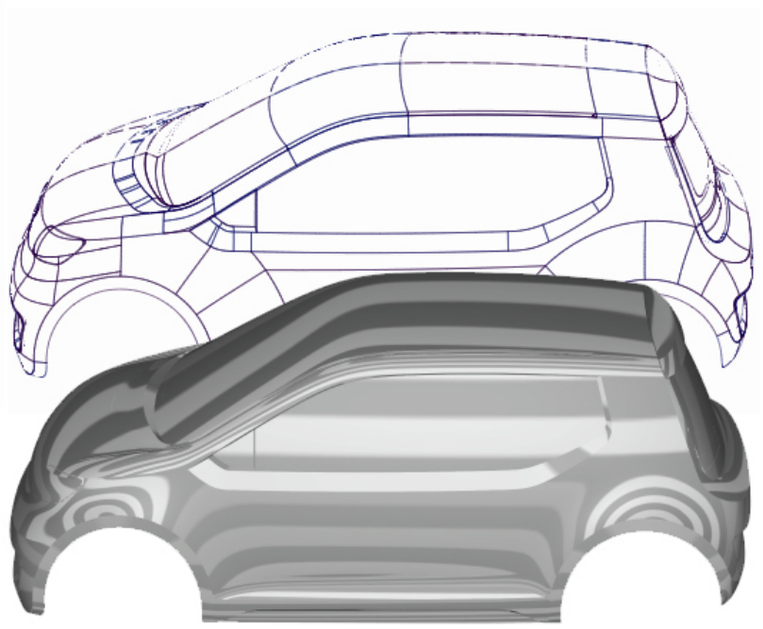


Surface

Polygon model

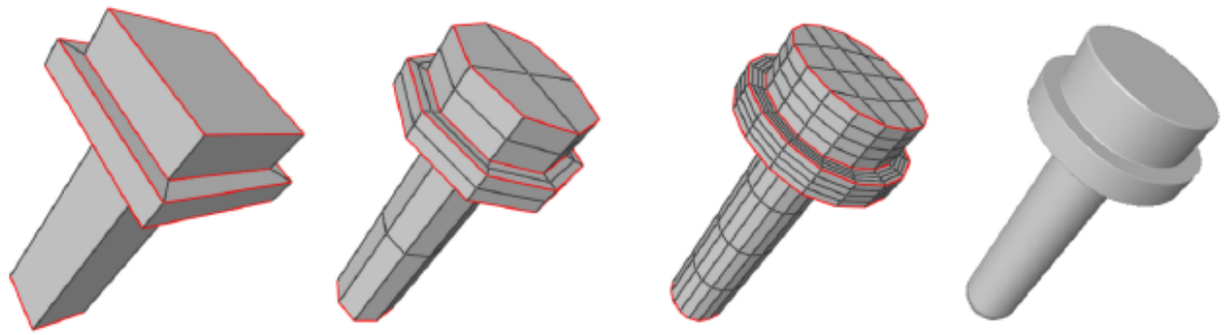


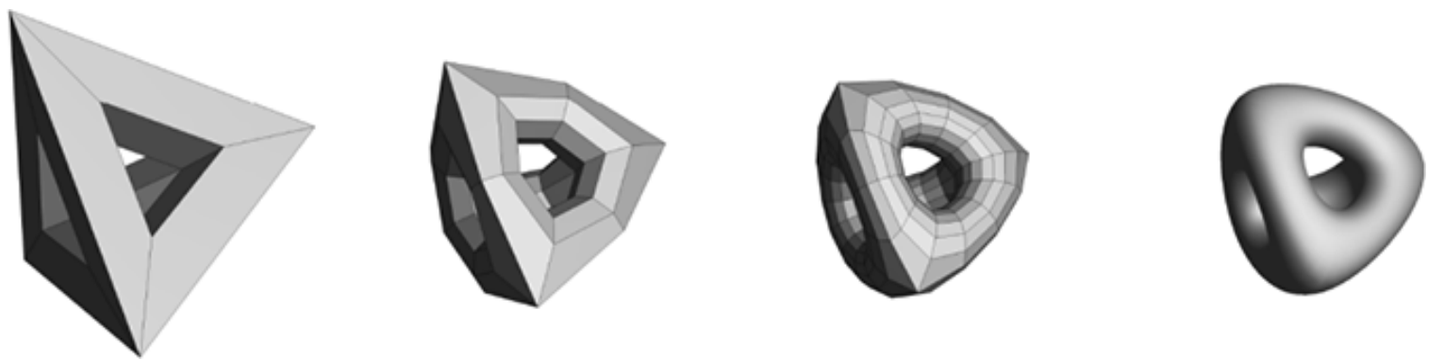
NURBS model (dérivable)



Modélisation géométrique > surfaces de subdivision

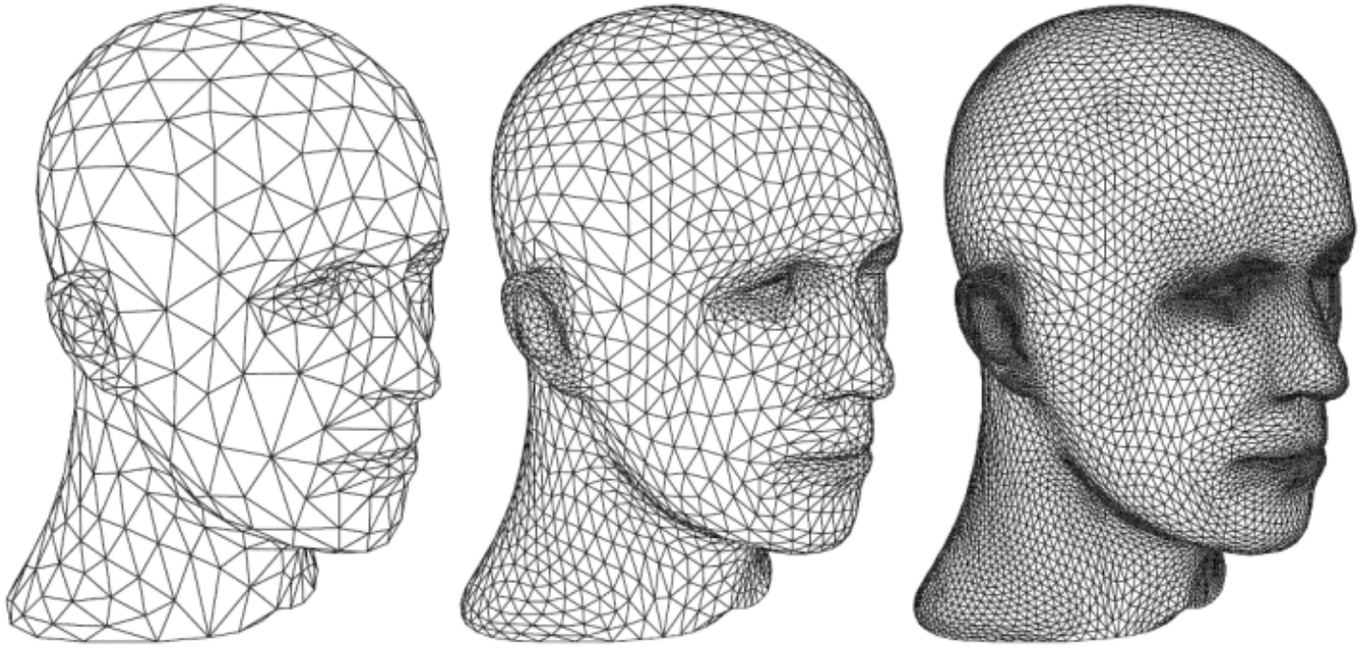
Subdivisions itératives des surfaces par introduction de nouveaux sommets calculés par interpolation des sommets présents à la précédente itération.





Le schéma d'interpolation retenu permet d'influer sur le résultat.

Pratique pour générer n'importe quel niveau de détail.
LOD-friendly.

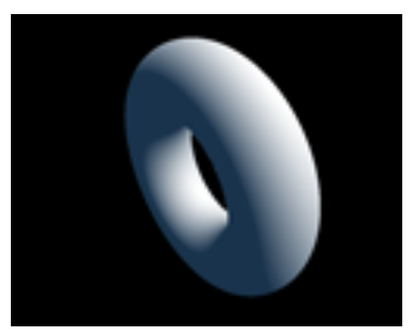


Modélisation procédurale > principe général

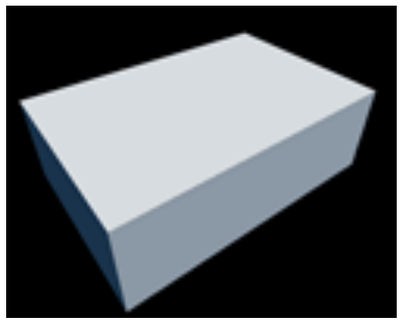
- Représenter « mathématiquement » les choses plutôt que géométriquement
    - *comprendre, « analytiquement »*
- Le modèle devient une formule pour calculer la distance à une forme
    - *représentation implicite versus explicite*
    - *calculs versus mémoire*
- Offre aussi quelques facilités pour différents problèmes
    - *objets « mous »*
    - *modèles très grands ou très complexes*
    - *détection de collisions (animation)*
    - *éclairage si dérivable (normale en tout point*

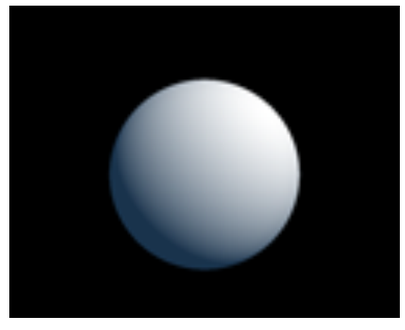
Modélisation procédurale > quelques primitives (quiz)

```
float f1( vec3 p, float s )
{
  return length(p)-s;
}
```



```
float f2( vec3 p, vec3 b )
{
  return length(max(abs(p)-b,0.0));
}
```

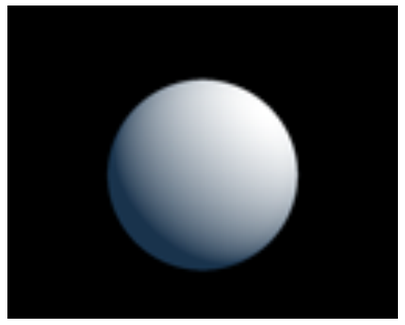




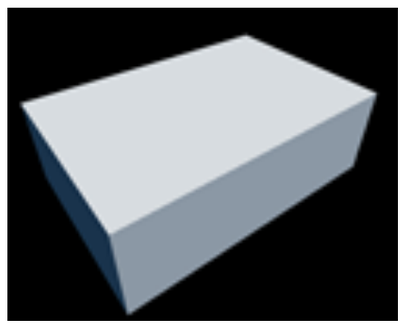
```
float f3( vec3 p, vec2 t )
{
  vec2 q = vec2(length(p.xz)-t.x,p.y);
  return length(q)-t.y;
}
```

Modélisation procédurale > quelques primitives

```
float sdSphere( vec3 p, float s )
{
  return length(p)-s;
}
```



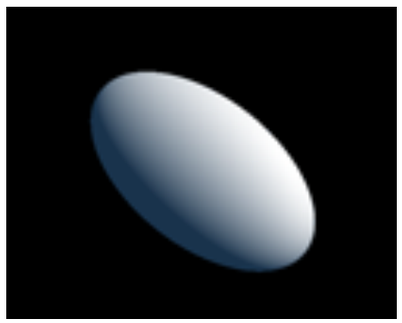
```
float udBox( vec3 p, vec3 b )
{
  return length(max(abs(p)-b,0.0));
}
```



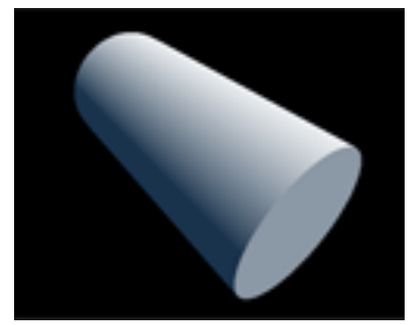
```
float sdTorus( vec3 p, vec2 t )
{
  vec2 q = vec2(length(p.xz)-t.x,p.y);
  return length(q)-t.y;
}
```

Modélisation procédurale > quelques primitives (quiz 2)

```
float f4( vec3 p, vec3 r )
{
  return (length( p/r ) - 1.0) *
          min(min(r.x,r.y),r.z);
}
```



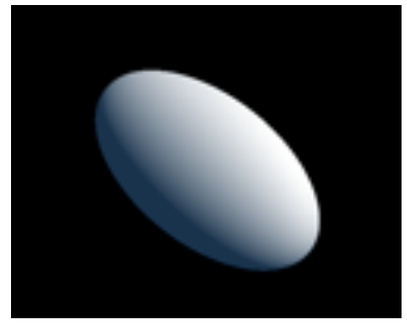
```
float f5( vec3 p, vec2 c )
{
    // c must be normalized
    float q = length(p.xy);
    return dot(c, vec2(q,p.z));
}
```



```
float f6( vec3 p, vec2 h )
{
  vec2 d = abs(vec2(length(p.xz),p.y)) - h;
  return min(max(d.x,d.y),0.0) +
         length(max(d,0.0));
}
```

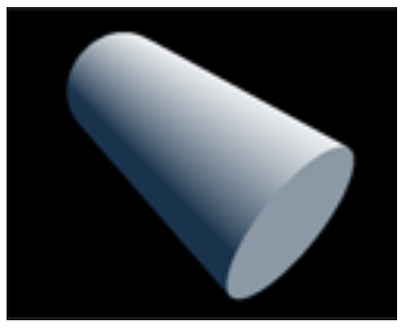
Modélisation procédurale > quelques primitives

```
float sdEllipsoid( vec3 p, vec3 r )
{
  return (length( p/r ) - 1.0) *
         min(min(r.x,r.y),r.z);
}
```



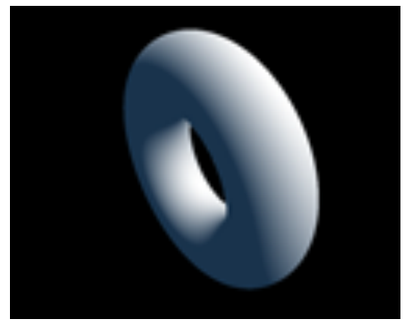
```
float sdCone( vec3 p, vec2 c )
{
    // c must be normalized
    float q = length(p.xy);
    return dot(c, vec2(q,p.z));
}
```

```
float sdCappedCylinder( vec3 p, vec2 h )
{
  vec2 d = abs(vec2(length(p.xz),p.y)) - h;
  return min(max(d.x,d.y),0.0) +
         length(max(d,0.0));
}
```



Modélisation procédurale > varier les distances

```
float sdTorus( vec3 p, vec2 t )
{
  vec2 q = vec2(length(p.xz)-t.x, p.y);
  return length(q)-t.y;
}
```



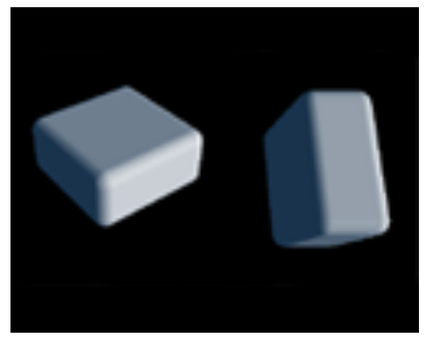
```
float sdTorus82( vec3 p, vec2 t )
{
  vec2 q = vec2(length(p.xz)-t.x, p.y);
  return length8(q)-t.y;
}
```

```
float sdTorus88( vec3 p, vec2 t )
{
  vec2 q = vec2(length8(p.xz)-t.x, p.y);
  return length8(q)-t.y;
}
```

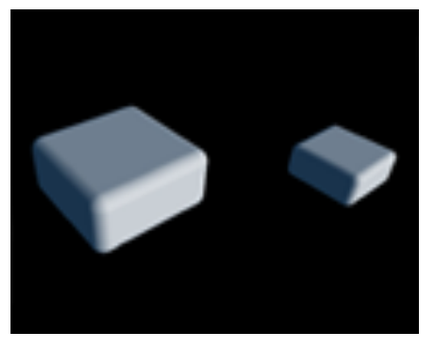
Modélisation procédurale > opérations sur les domaines

```
float opRep( vec3 p, vec3 c )
{
    vec3 q = mod(p,c)-0.5*c;
    return primitive(q);
}
```

```
vec3 opTx( vec3 p, mat4 m )
{
    vec3 q = invert(m)*p;
    return primitive(q);
}
```

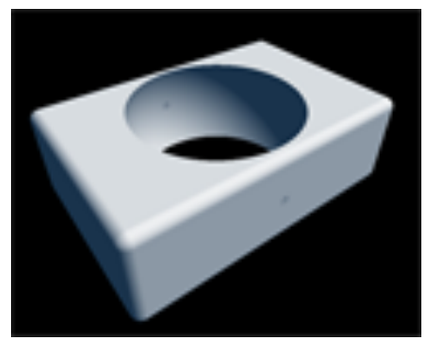


```
float opScale( vec3 p, float s )
{
    return primitive(p/s)*s;
}
```

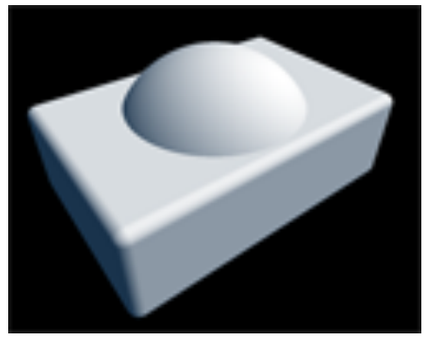


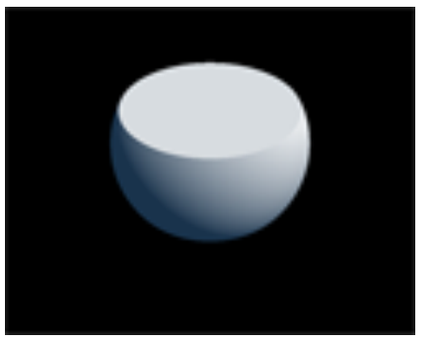
Modélisation procédurale > opérations (quiz 3)

```
float op1( float d1, float d2 )
{
    return min(d1,d2);
}
```



```
float op2( float d1, float d2 )
{
    return max(-d1,d2);
}
```

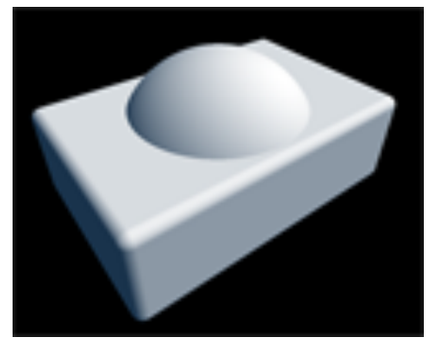




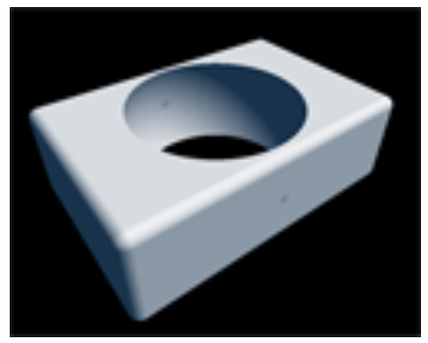
```
float op3( float d1, float d2 )
{
    return max(d1,d2);
}
```

Modélisation procédurale > opérations (quiz 3)

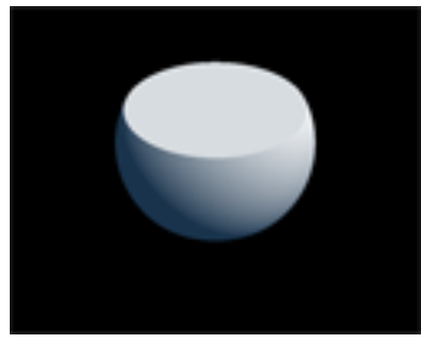
```
float opUnion( float d1, float d2 )
{
    return min(d1,d2);
}
```



```
float opSubstraction( float d1, float d2 )
{
    return max(-d1,d2);
}
```



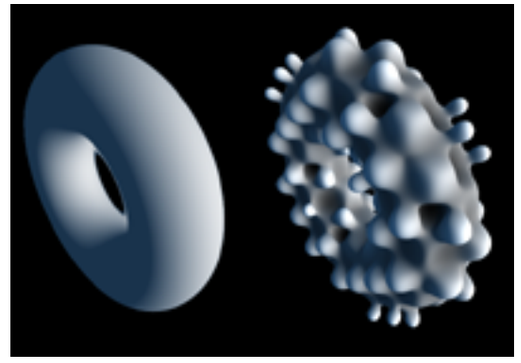
```
float opIntersection( float d1, float d2 )
{
    return max(d1,d2);
}
```



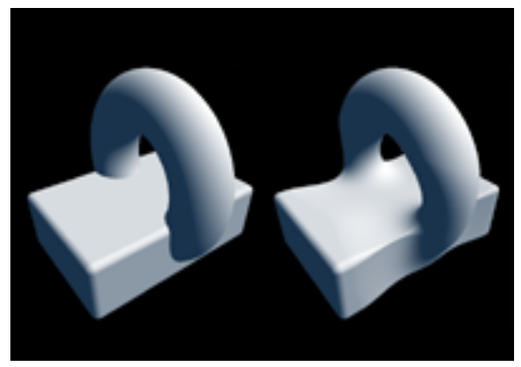
Modélisation procédurale > déformer les distances

```
float displacement( vec3 p )
{
    return sin(20.0*p.x) +
           sin(20.0*p.y) + sin(20.0*p.z);
}

float opDisplace( vec3 p )
{
    float d1 = primitive(p);
    float d2 = displacement(p);
    return d1+d2;
}
```

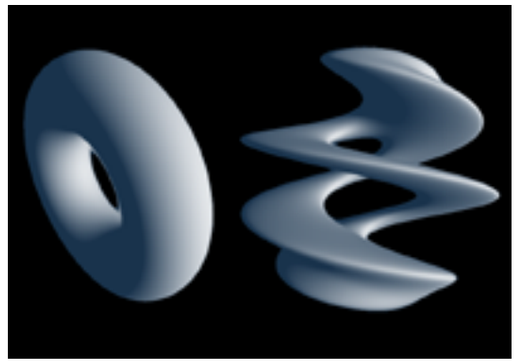


```
float opBlend( vec3 p )
{
    float d1 = primitiveA(p);
    float d2 = primitiveB(p);
    return smin( d1, d2 );
}
```

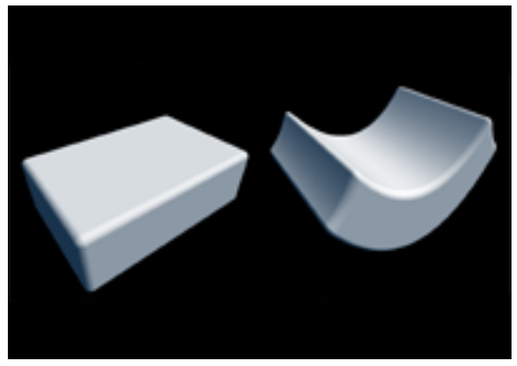


Modélisation procédurale > déformer les domaines

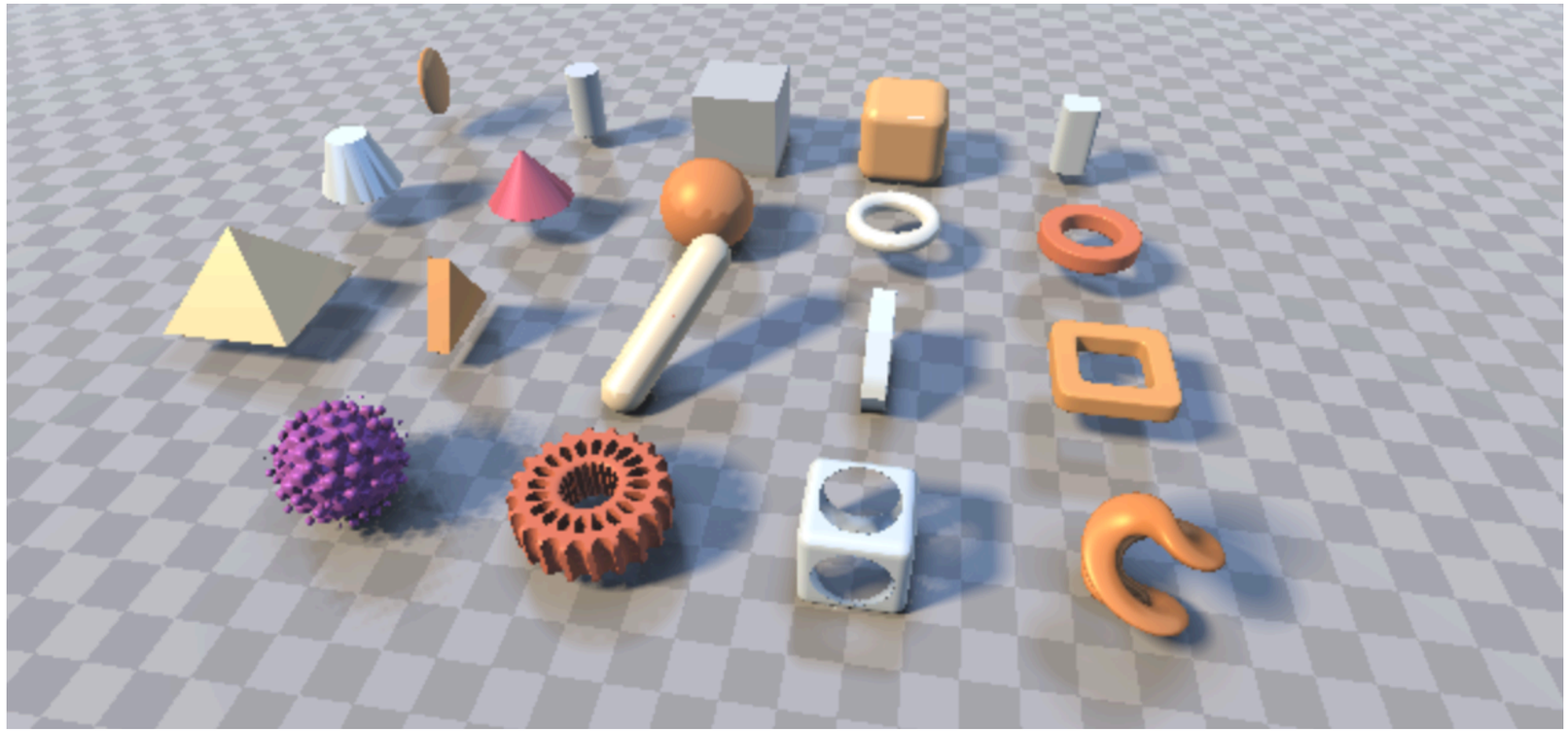
```
float opTwist( vec3 p )
{
    float c = cos(20.0*p.y);
    float s = sin(20.0*p.y);
    mat2  m = mat2(c,-s,s,c);
    vec3  q = vec3(m*p.xz,p.y);
    return primitive(q);
}
```



```
float opCheapBend( vec3 p )
{
    float c = cos(20.0*p.y);
    float s = sin(20.0*p.y);
    mat2  m = mat2(c,-s,s,c);
    vec3  q = vec3(m*p.xy,p.z);
    return primitive(q);
}
```

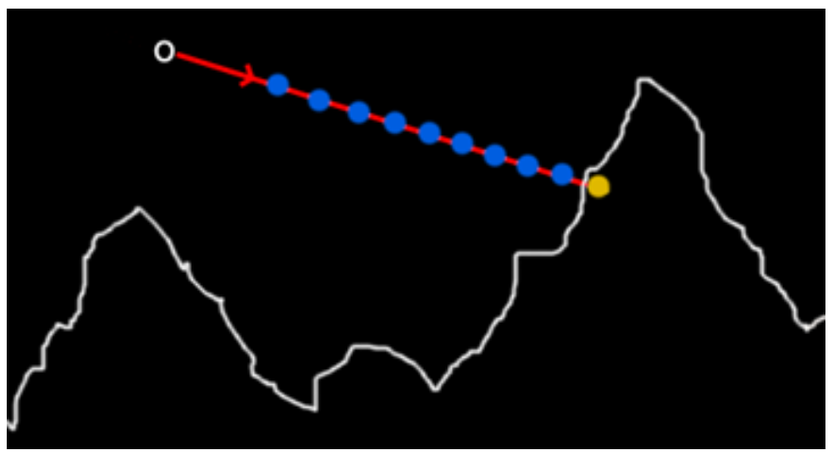


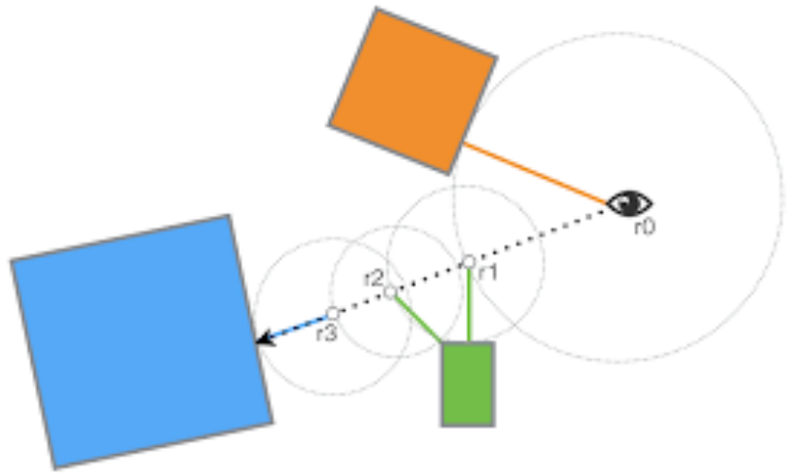
Modélisation procédurale > example



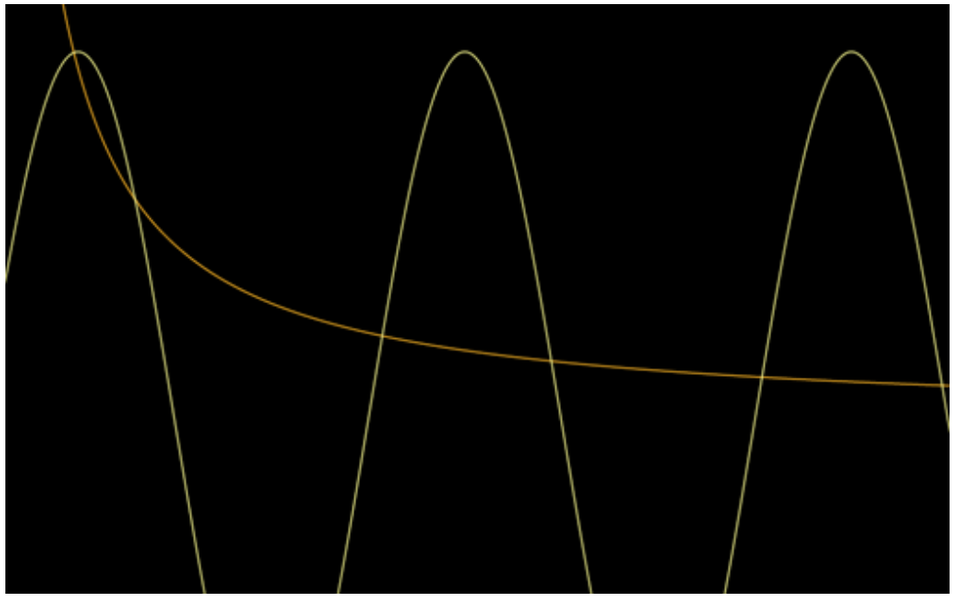
Au passage : comment on rend cela ?

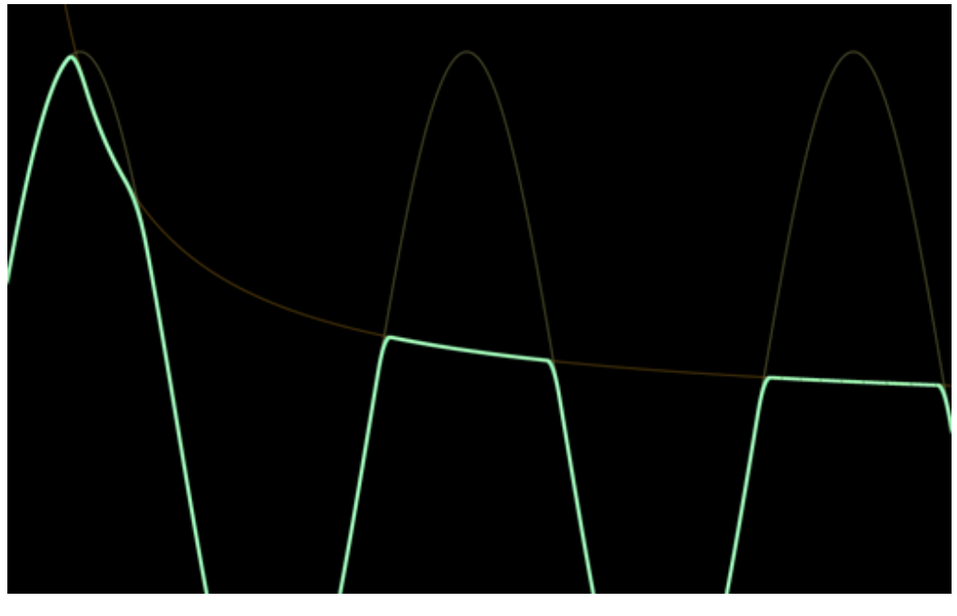
- Ray marching -





Modélisation procédurale > « smooth minimum »

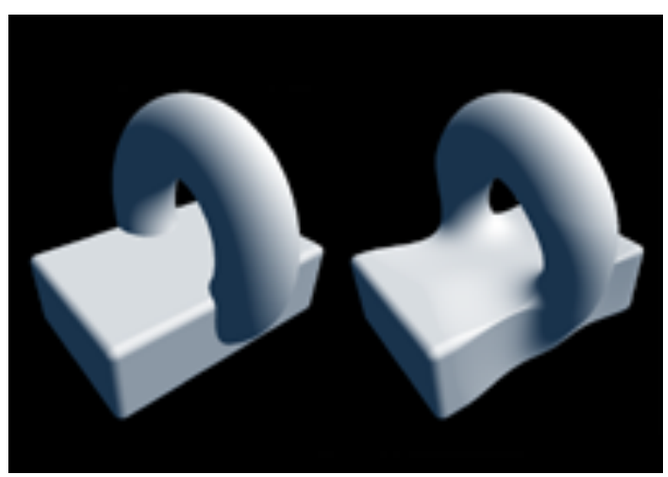




clamp : limiter un calcul à un intervalle donné, mix : interpolation linéaire

```
float smin( float a, float b, float k )
{
    float h = clamp( 0.5+0.5*(b-a)/k, 0.0, 1.0 );
    return mix( b, a, h ) - k*h*(1.0-h);
}

float opBlend( vec3 p )
{
    float d1 = primitiveA(p);
    float d2 = primitiveB(p);
    return smin( d1, d2 );
}
```



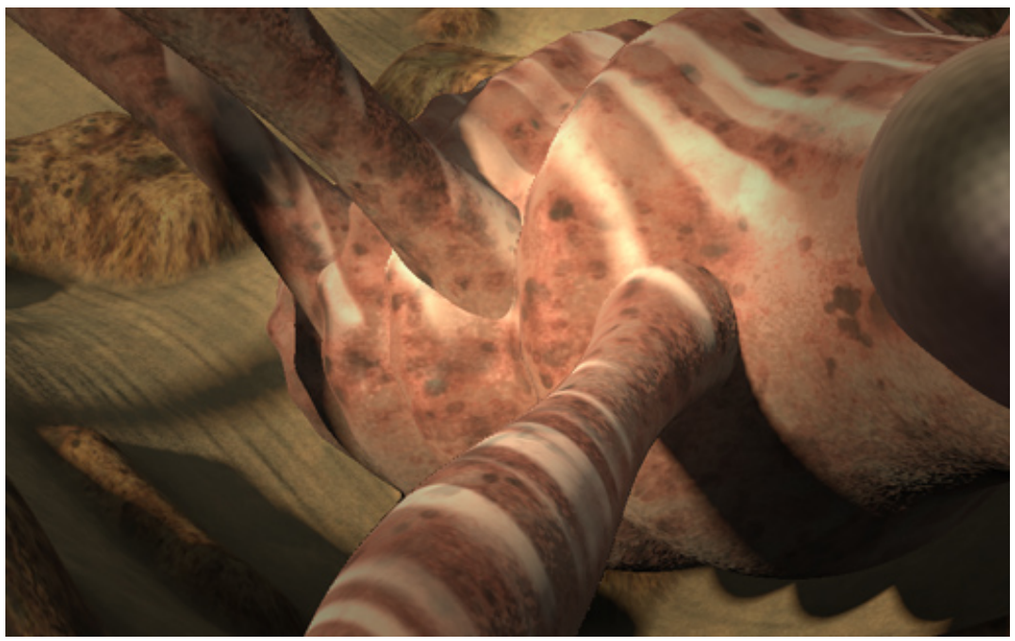
LiveSlides web content

To view

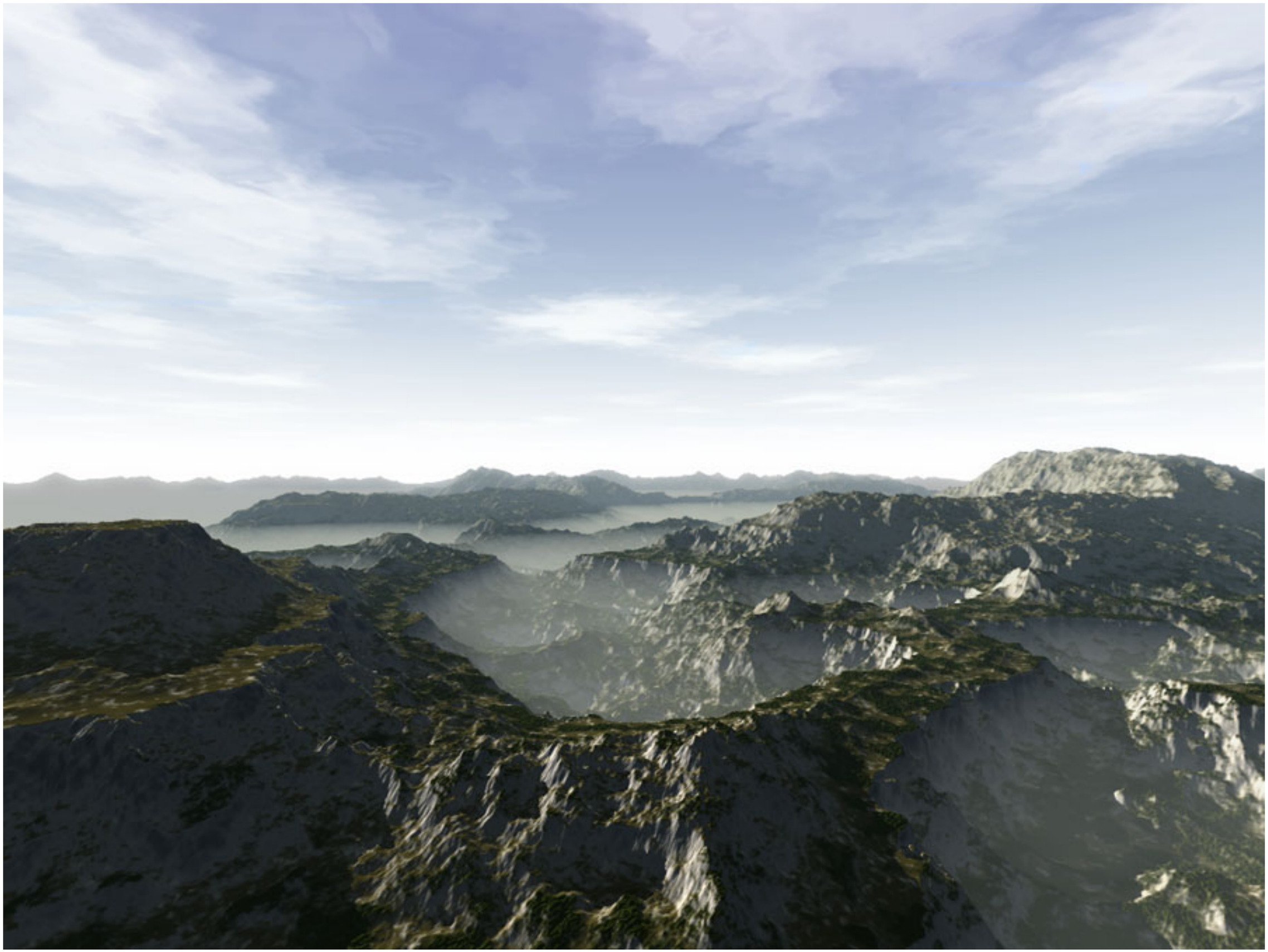
Download the add-in.
liveslides.com/download

Start the presentation.

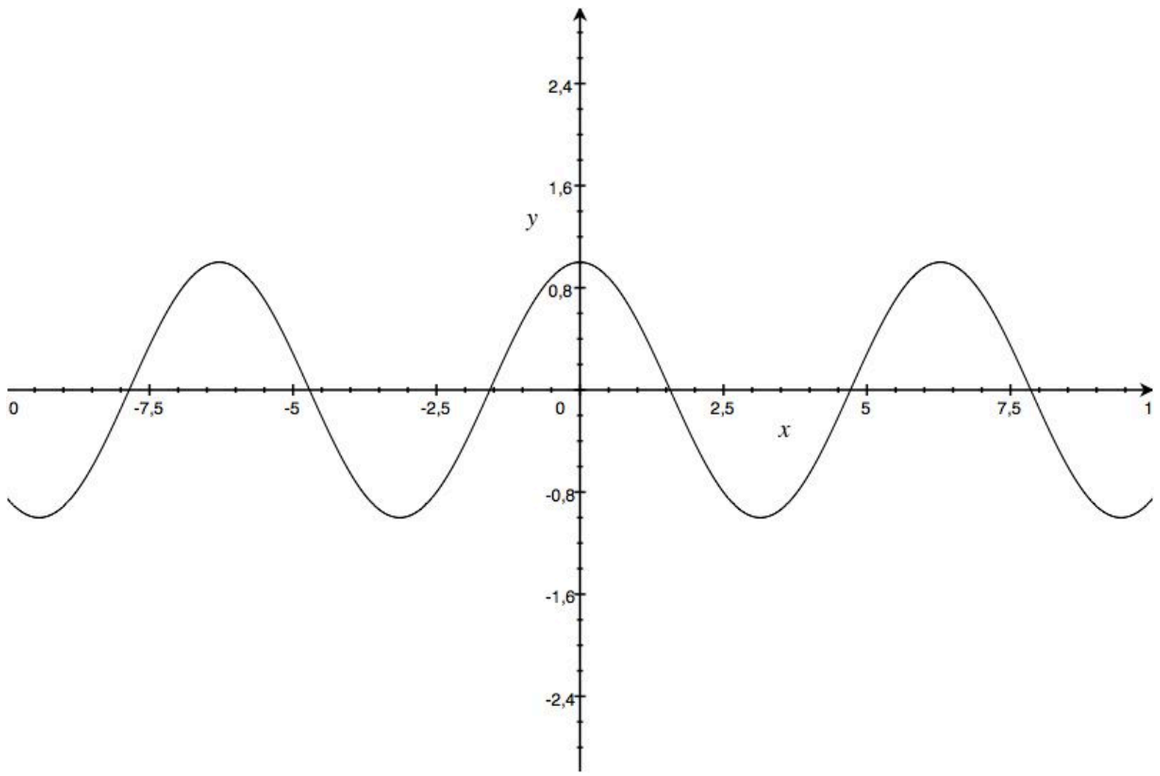
Modélisation procédurale > examples

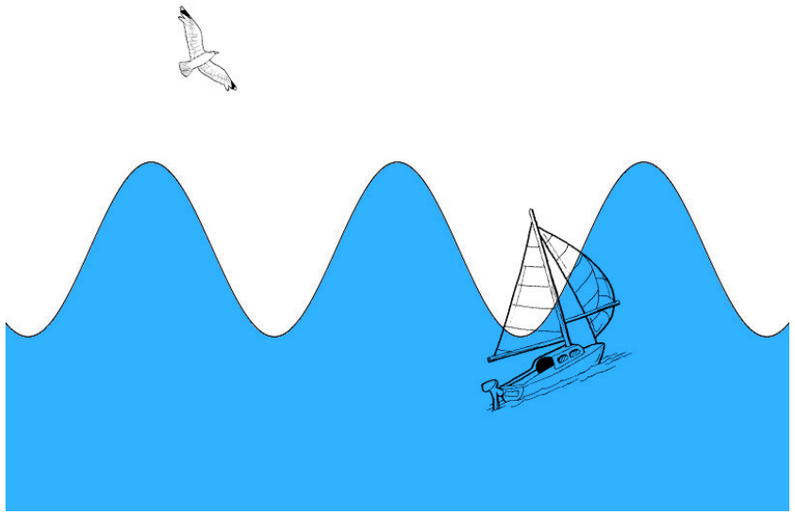


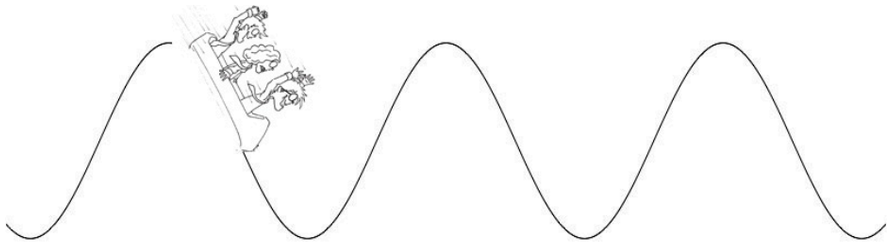
Modélisation procédurale > techniques (plus) avancées

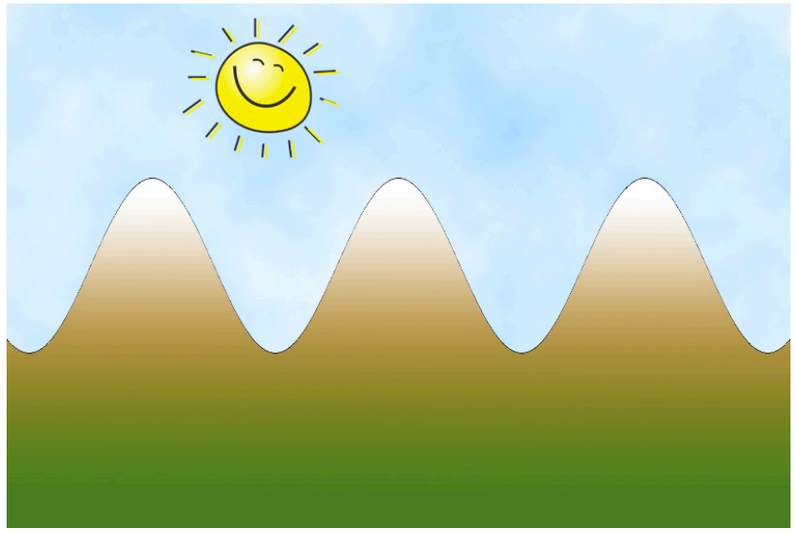


Modélisation procédurale > techniques (plus) avancées









Le syndrome du parquet stratifié

Modélisation procédurale > fractional Brownian motion

Le **mouvement Brownien fractionnaire** (mBf) a été introduit par Kolmogorov en 1940 comme moyen d'engendrer des "spirales" gaussiennes dans des espaces de Hilbert.

Mandelbrot et Van Ness (1968) l'ont rendu célèbre en l'introduisant dans des modèles financiers et en étudiant ses propriétés. Le champ des applications du mBf est immense. En effet, il sert par exemple à recréer certains paysages naturels, notamment des montagnes, mais également en hydrologie, télécommunications, économie, physique...

Le mouvement Brownien fractionnaire d'exposant de Hurst $\alpha \in (0,1)$, noté $\{B_\alpha(t)\}_{t\in\mathbb{R}}$, est l'unique processus Gaussien centré, nul en zéro et continu dont la covariance est donnée par:

$$\mathbb{E}(B_\alpha(s)B_\alpha(t)) = \frac{C_\alpha}{2}(|s|^{2\alpha} + |t|^{2\alpha} - |s-t|^{2\alpha}),$$

où $C_\alpha = \mathrm{Var}(\mathrm{B}_\alpha(1))$ est une constante positive qui ne dépend que de α, elle s'appelle indice de Hurst. Lorsque $C_\alpha = 1$, nous obtenons le mBf standard. Le mBf est l'une des généralisations les plus naturelles du mouvement brownien. En effet,

1. Lorsque $\alpha > 1/2$, B_α est une primitive fractionnaire du mouvement brownien.
2. lorsque $\alpha < 1/2$, il est une dérivée fractionnaire du mouvement brownien.
3. $B_{1/2}$ se réduit à un mouvement brownien.

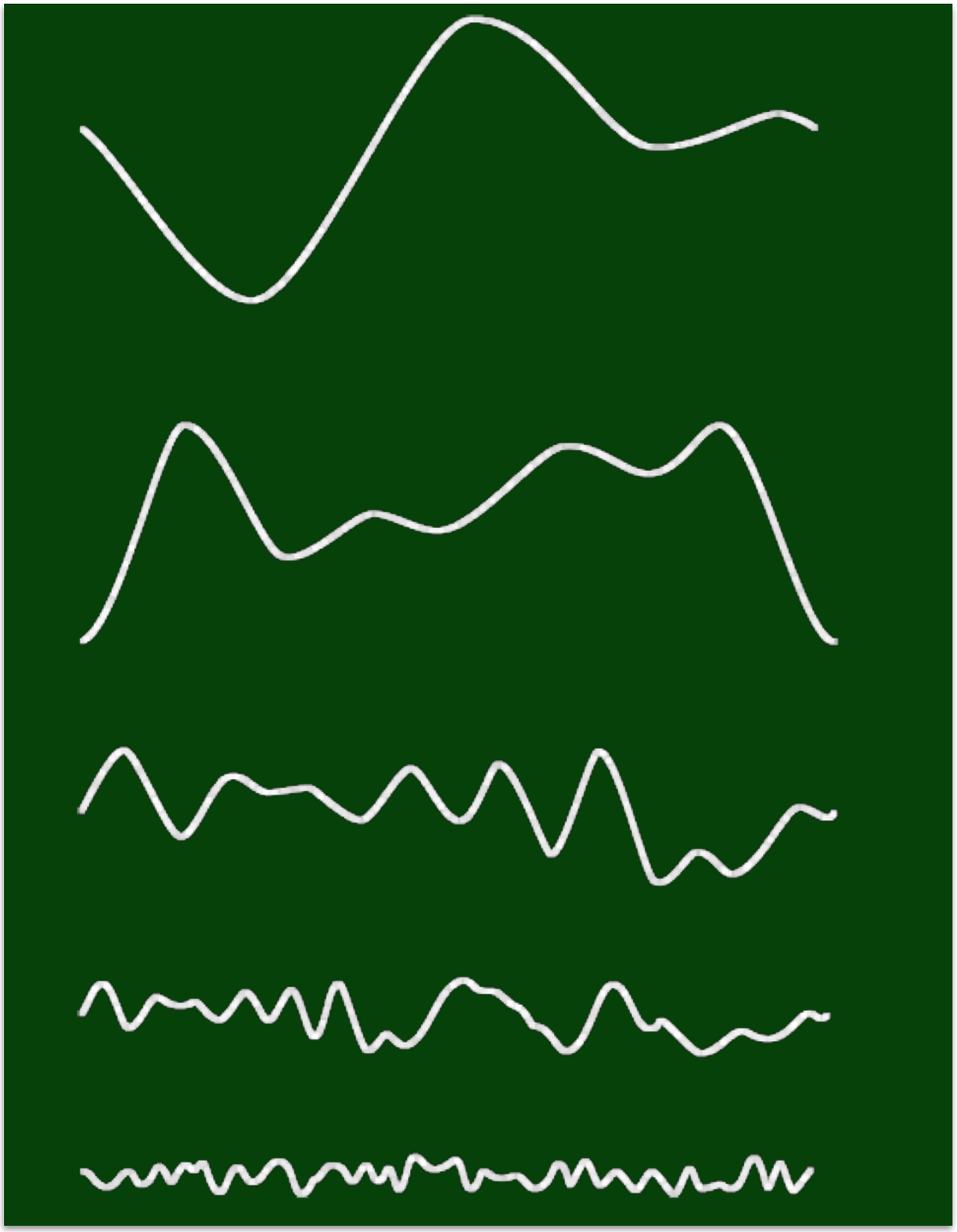
WIKIPEDIA
The Free Encyclopedia

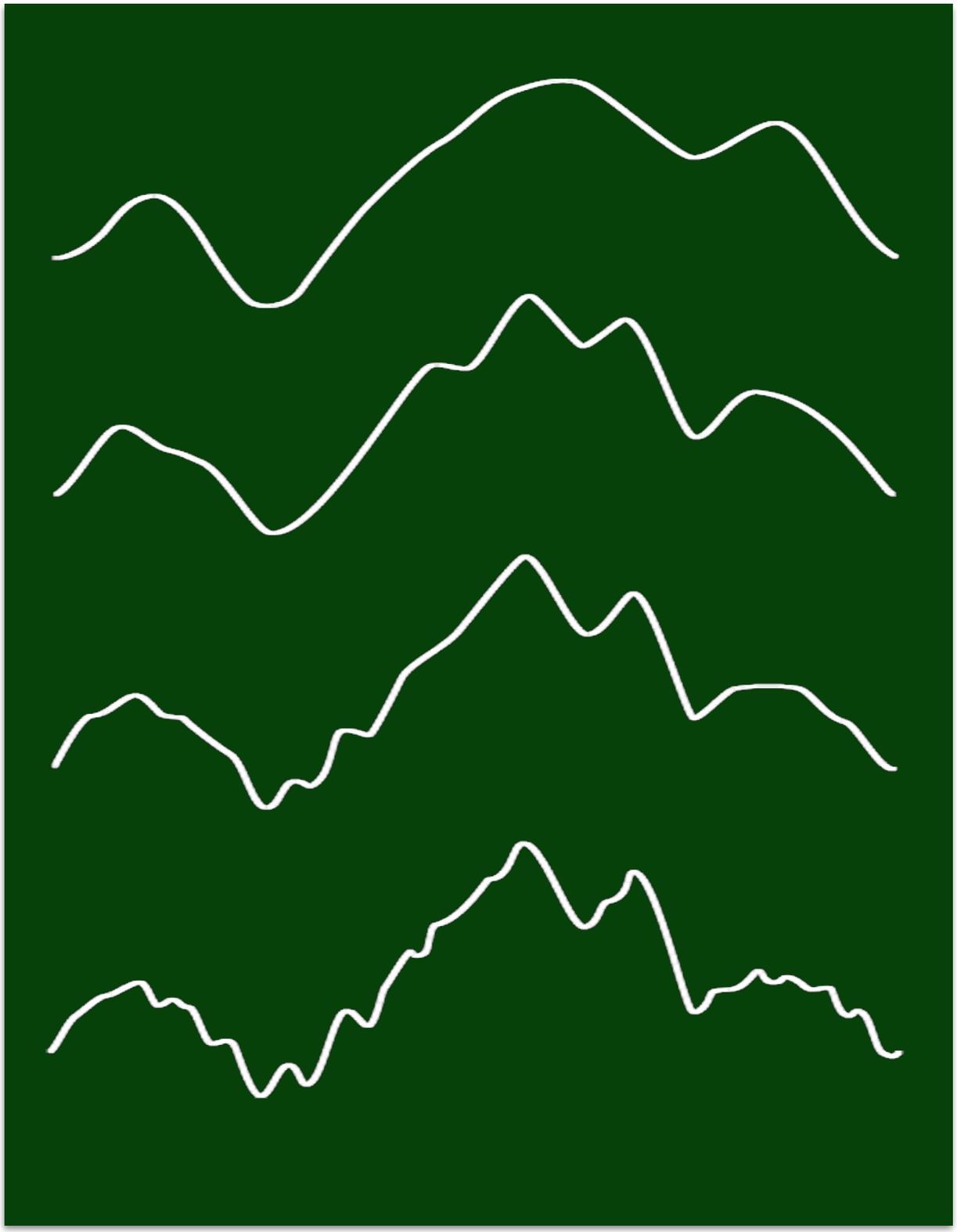
Modélisation procédurale > fractional Brownian motion

- Concrètement : une somme de fonctions de bruit (de Perlin souvent)
    - *masquer les aspects périodiques*

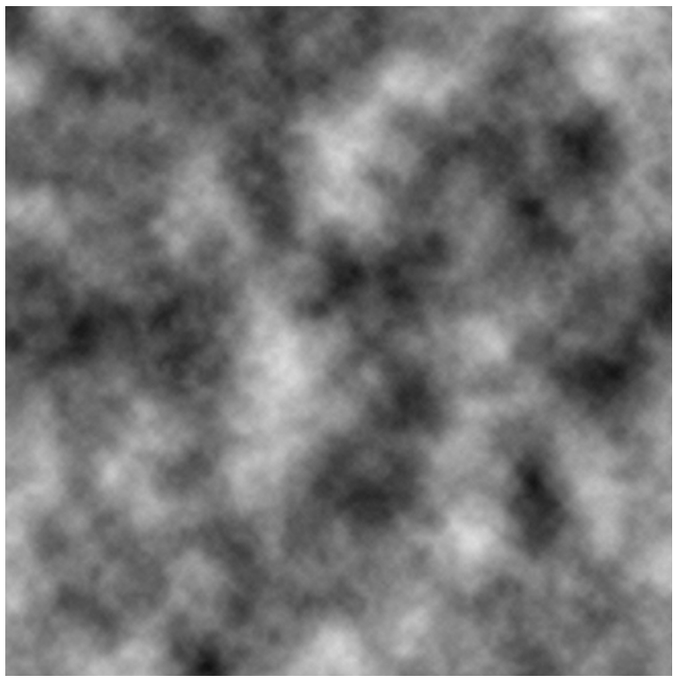
- fonctions de fréquences croissantes et d’amplitudes décroissantes
    - *fréquence basse, amplitude forte : forme globale*
    - *fréquence haute, amplitude faible : ajout de détails*

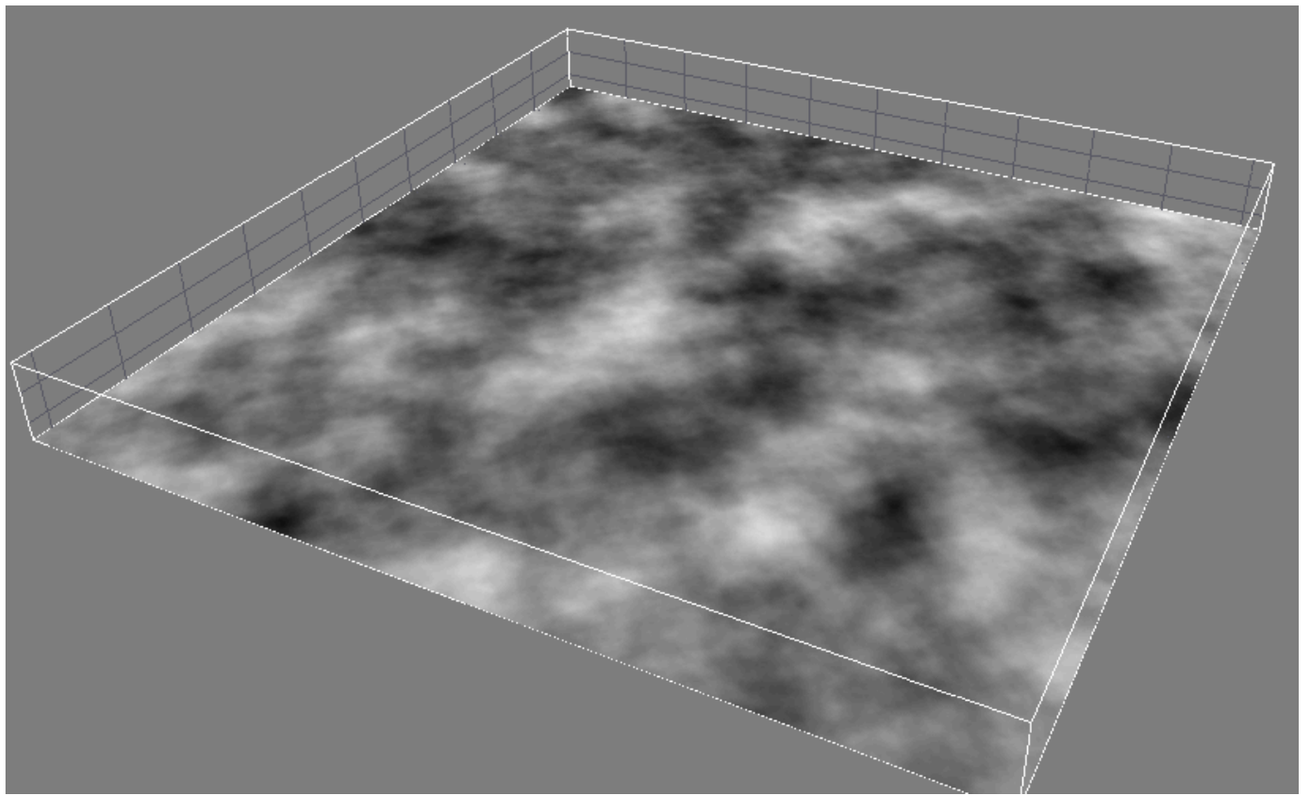
Modélisation procédurale > fractional Brownian motion

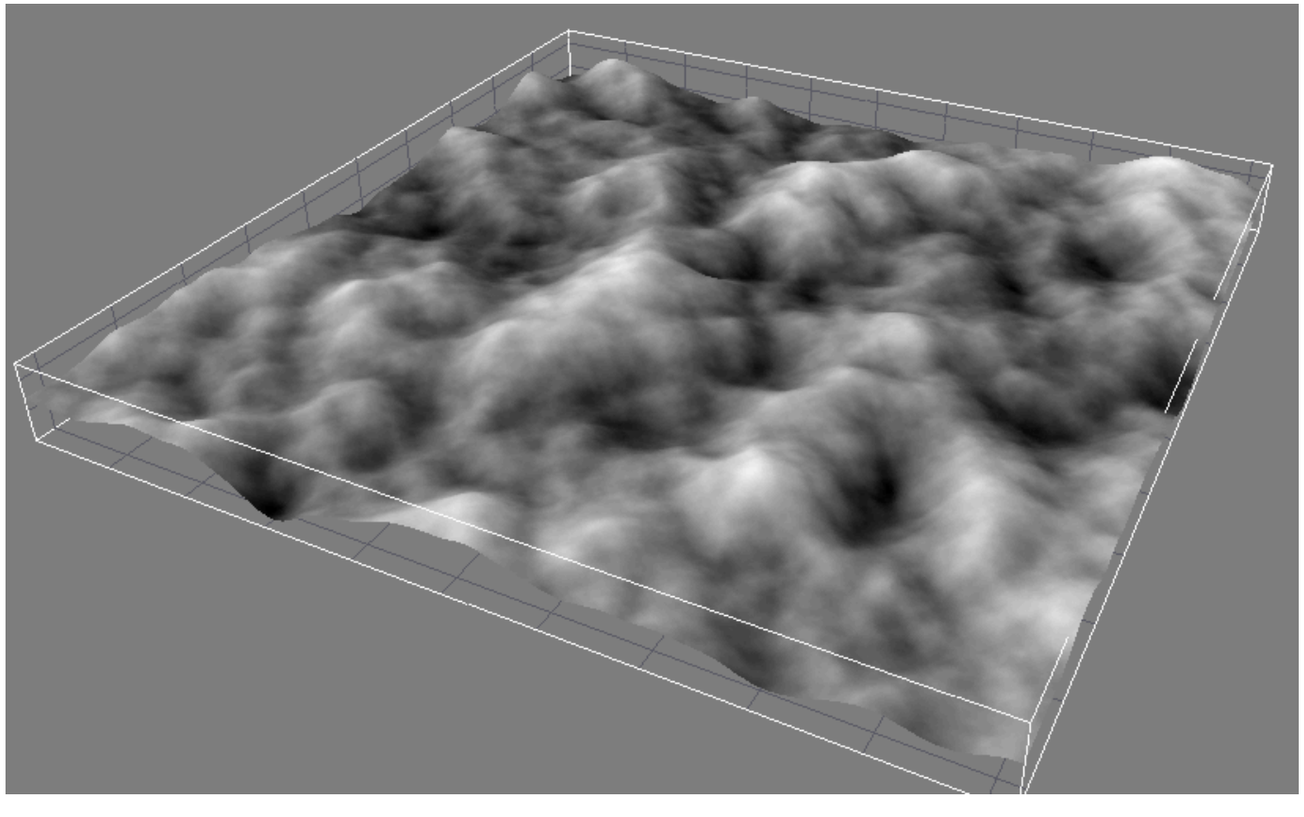




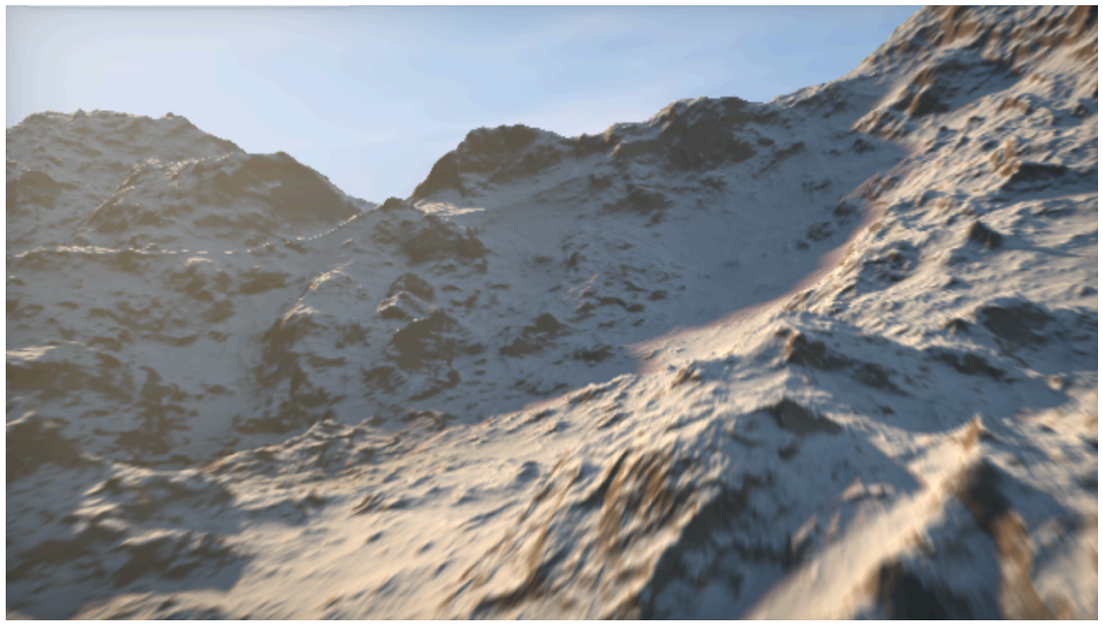
Modélisation procédurale > fractional Brownian motion

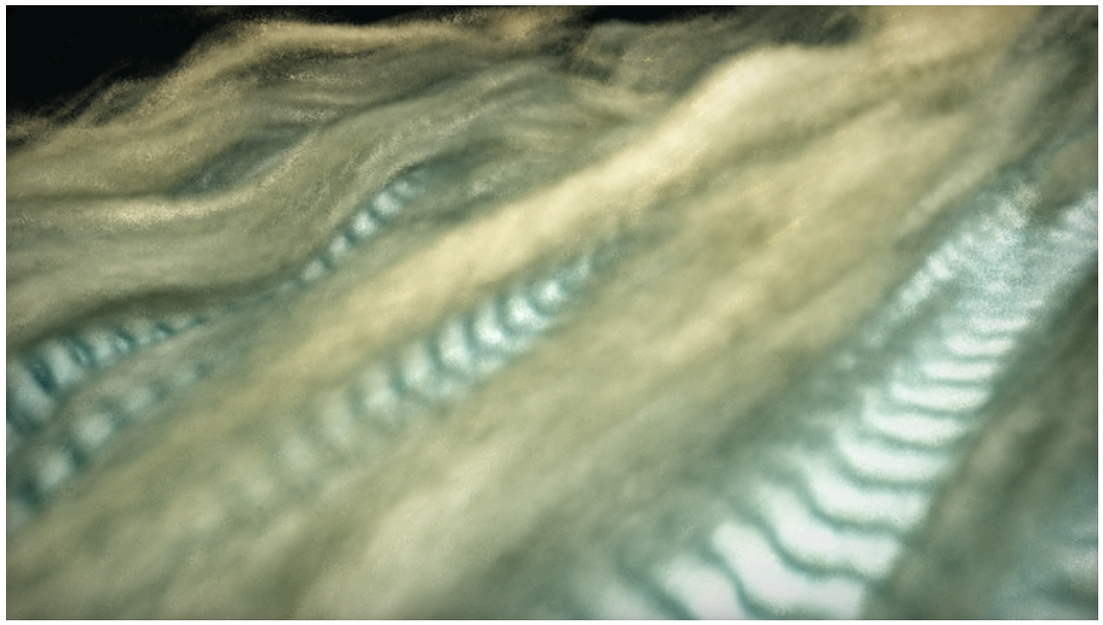




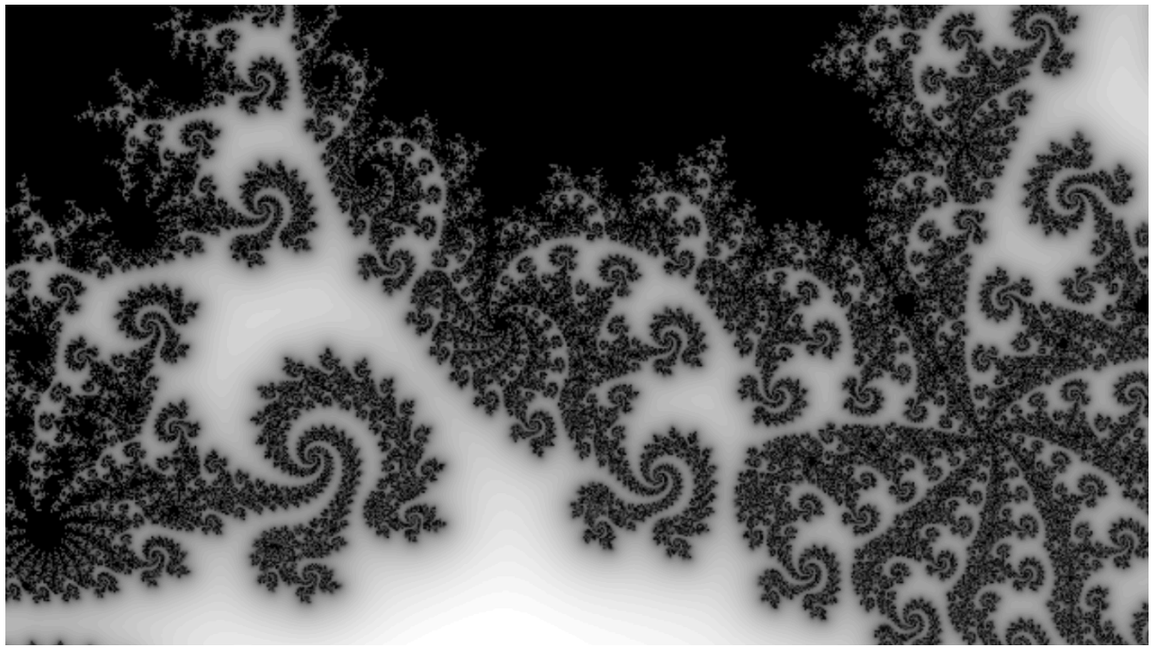


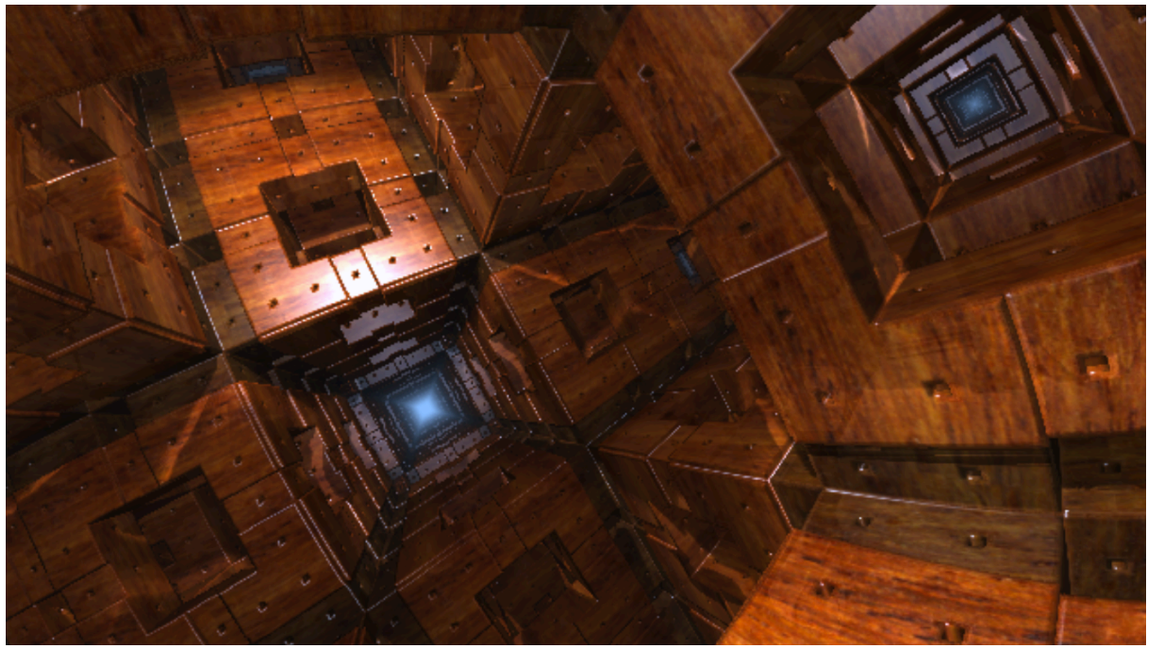
Modélisation procédurale > fractional Brownian motion

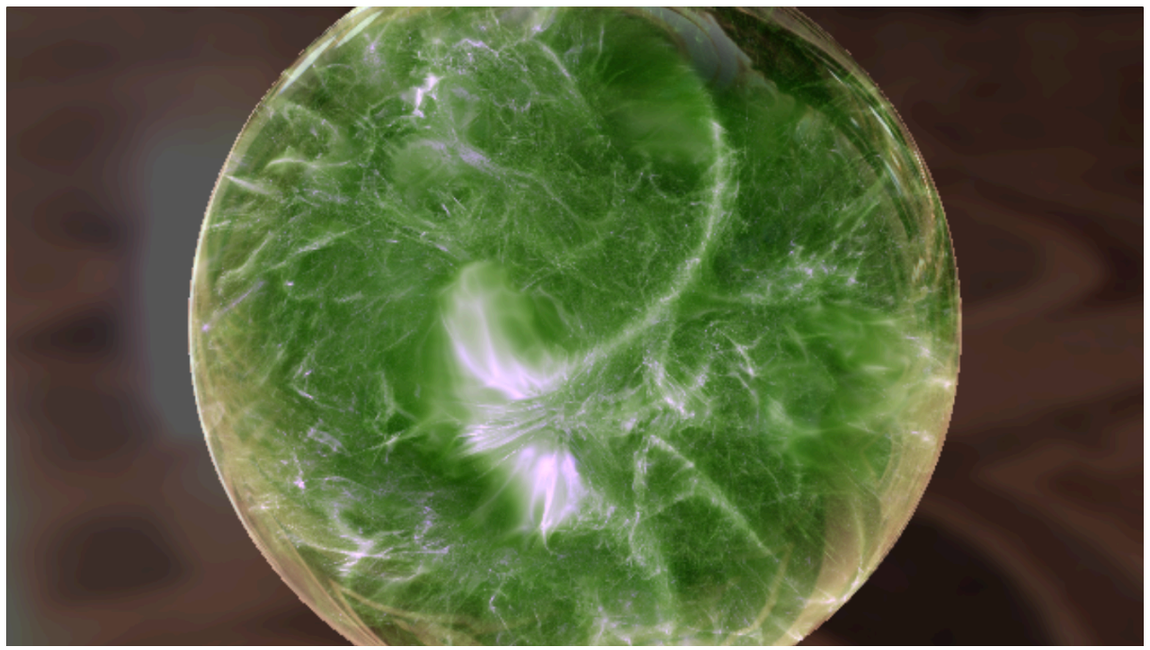


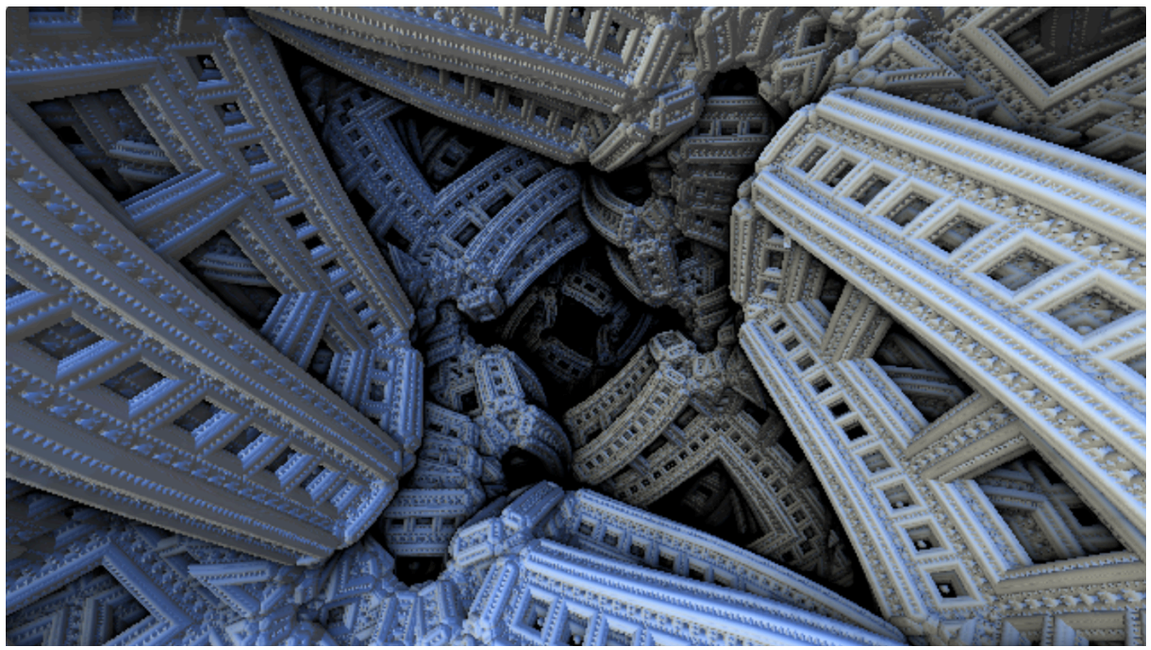


Modélisation procédurale > bonus : les fractales

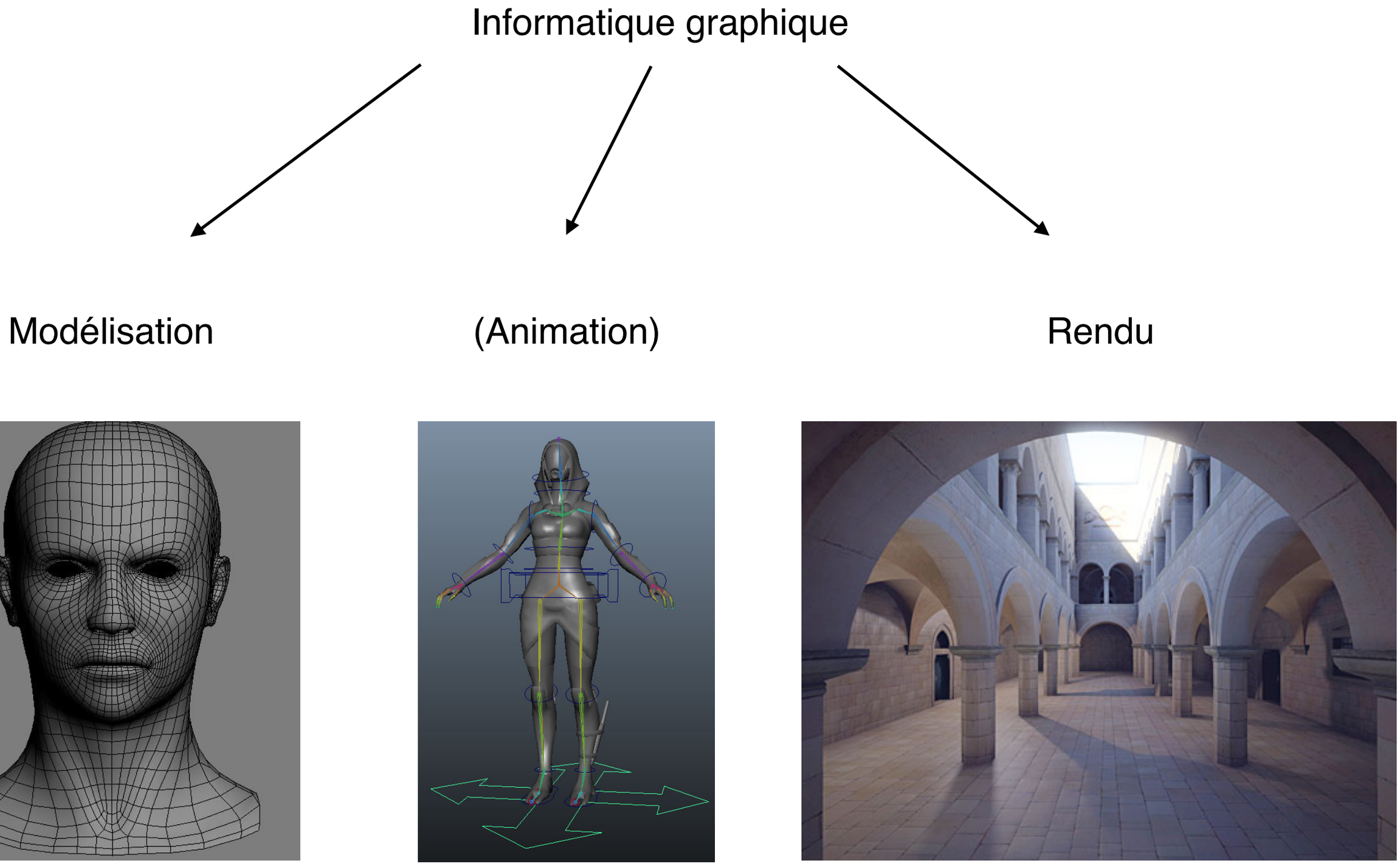




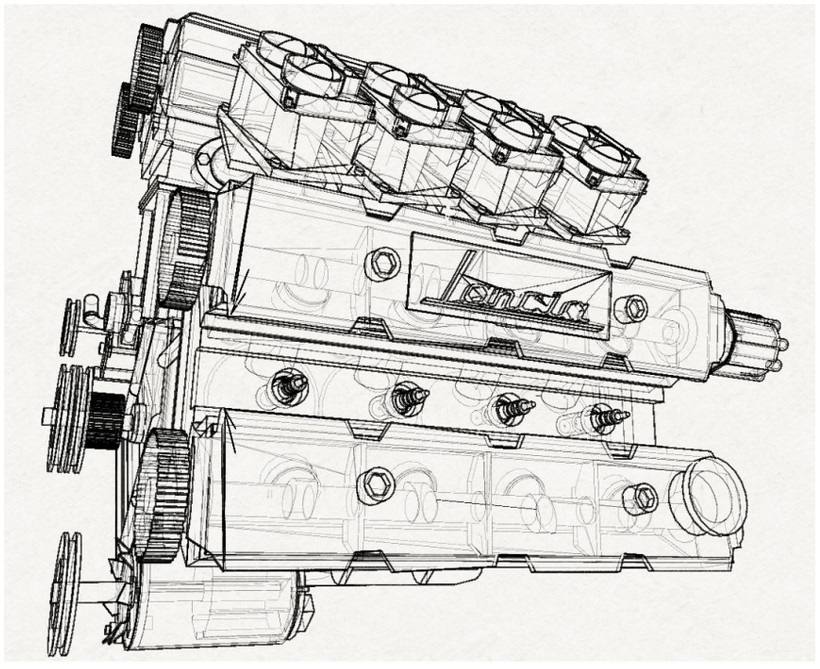


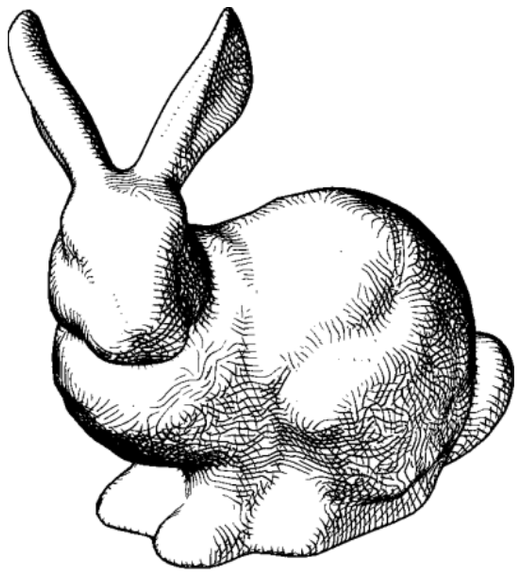


Informatique Graphique > rendu

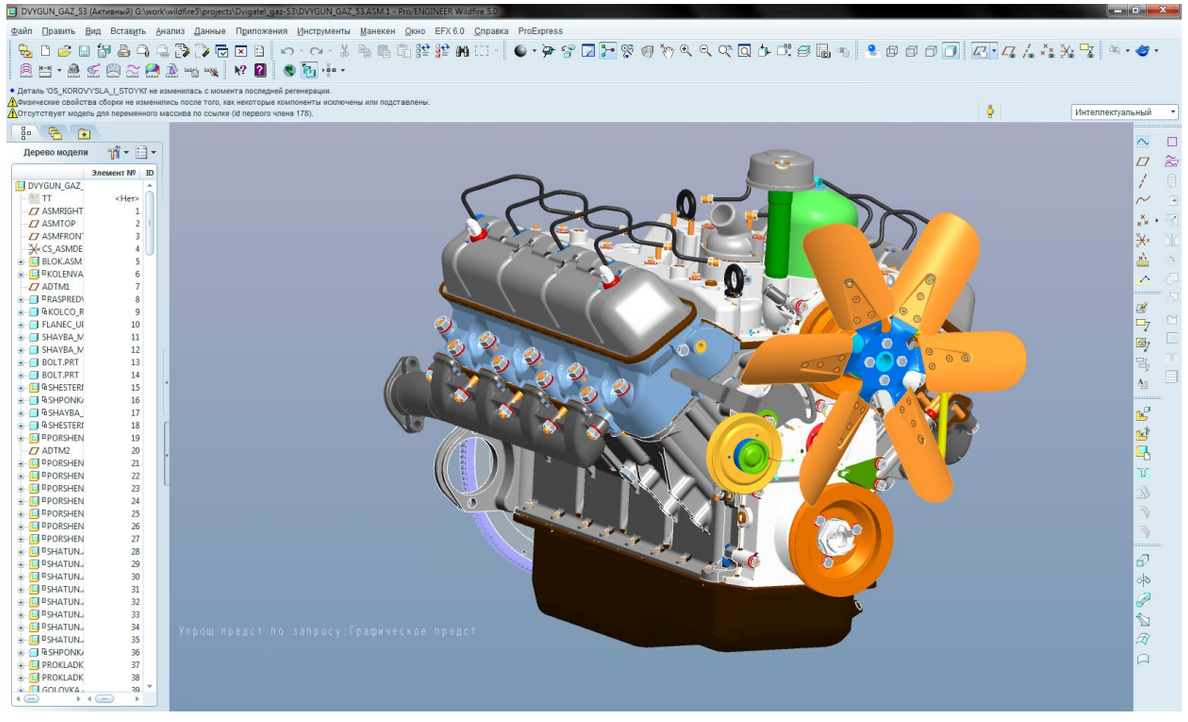


Rendu > Non Photo-Réaliste (NPR)





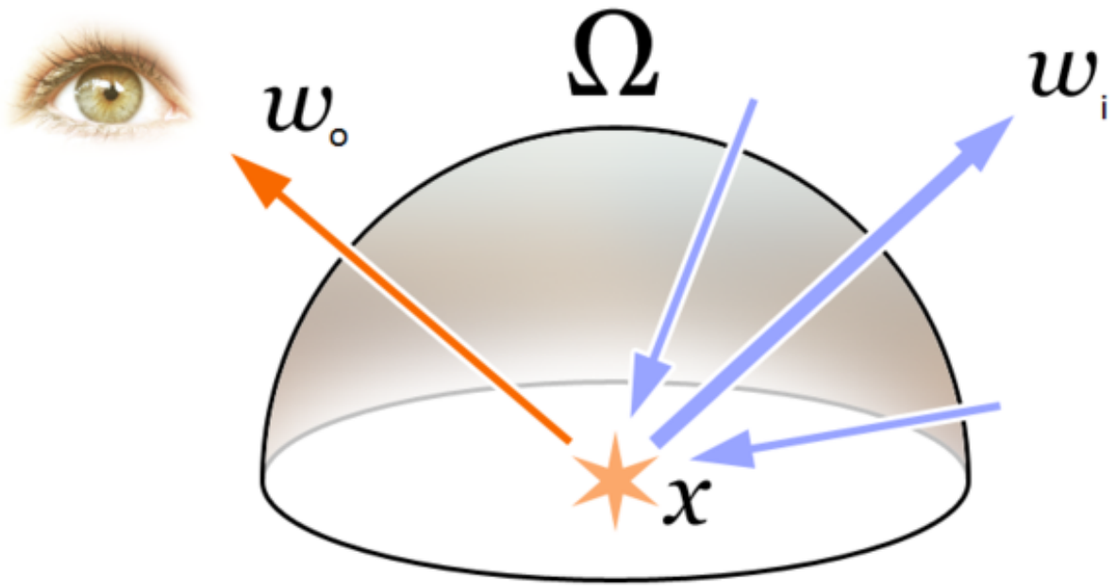
Rendu > Temps réel (60 images par seconde, 0.016s par image…)



Rendu > Réaliste (à base physique)

Rendu réaliste > l'équation de rendu [Kajiya 1986]

- Un résultat inspiré de la thermique
- Décrit le comportement de la lumière dans un environnement 3D

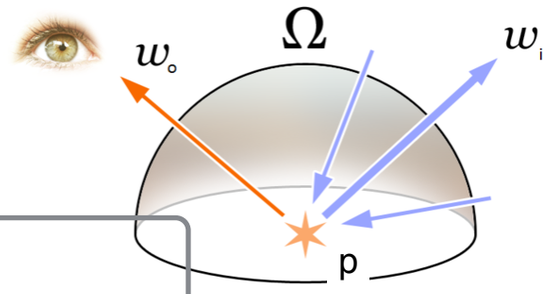


outgoing/observed radiance — emitted radiance (e.g., light source) — angle between incoming direction and normal — incoming radiance

$$L_o(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_{\mathcal{H}^2} f_r(\mathbf{p}, \omega_i \rightarrow \omega_o) L_i(\mathbf{p}, \omega_i) \cos\theta \, d\omega_i$$

point of interest — direction of interest — all directions in hemisphere — scattering function — incoming direction

Rendu réaliste > l'équation de rendu [Kajiya 1986]



0 si ce n'est pas une source

et toujours le normale

outgoing/observed radiance

emitted radiance (e.g., light source)

incoming radiance

angle between incoming direction and normal

$$L_o(\mathbf{p},\omega_o) = L_e(\mathbf{p},\omega_o) + \int_{\mathcal{H}^2} f_r(\mathbf{p},\omega_i \rightarrow \omega_o) L_i(\mathbf{p},\omega_i) \cos\theta \, d\omega_i$$

point of interest

direction of interest

all directions in hemisphere

scattering function

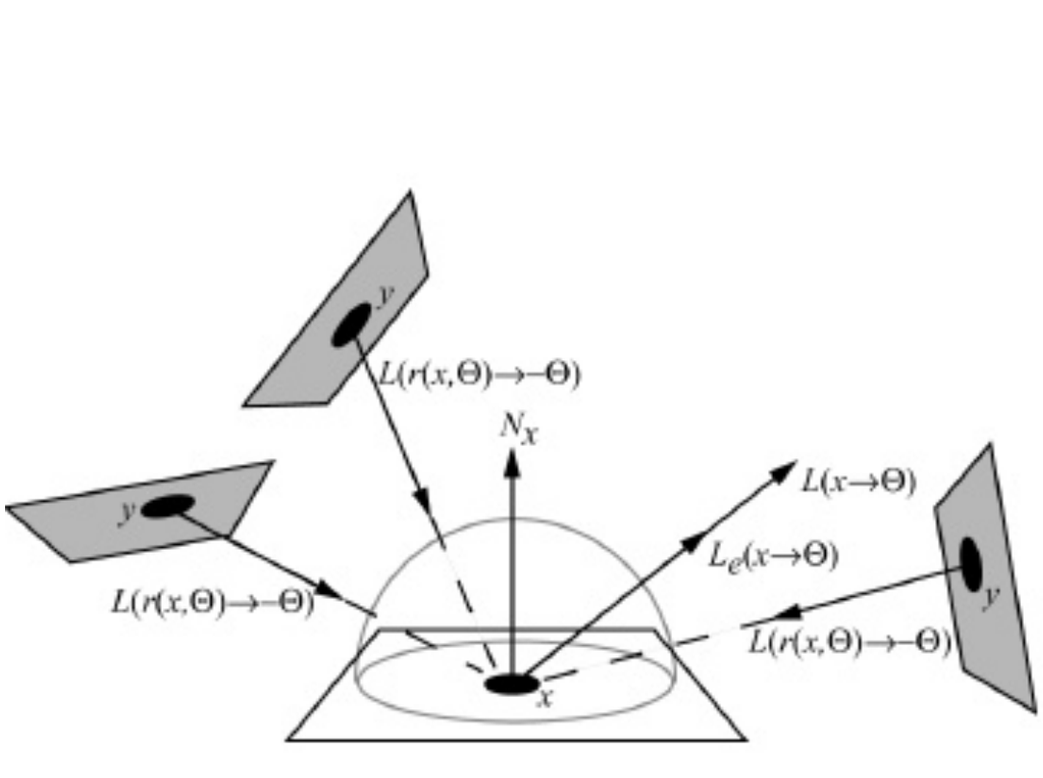
incoming direction

BRDF : Bidirectional Reflectance Distribution Function

Equation récursive !

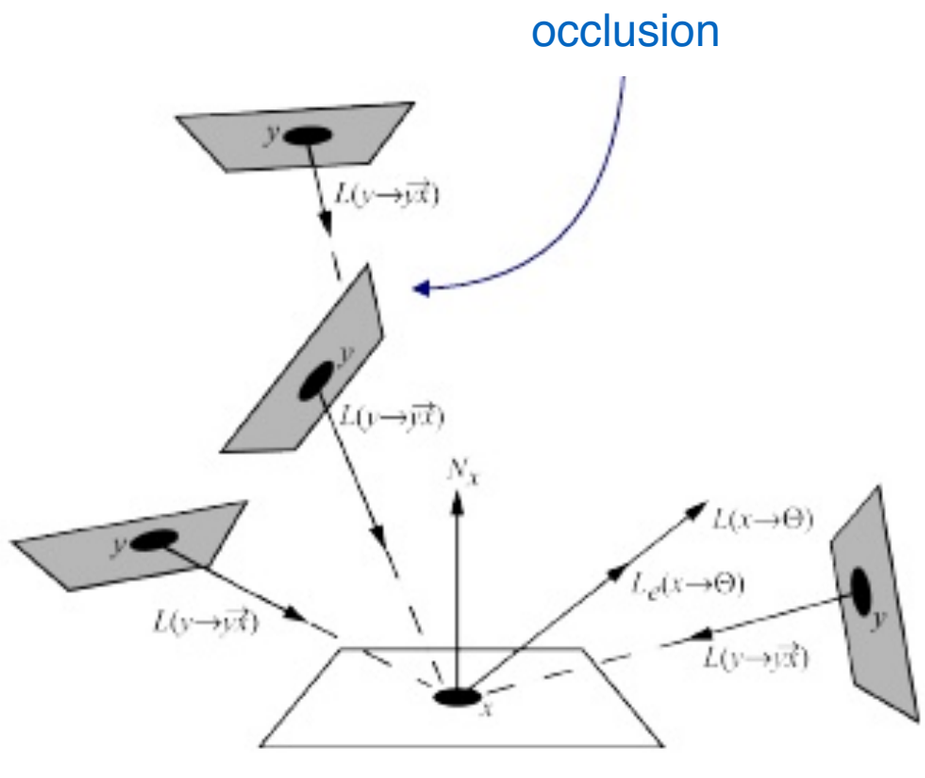
Rendu réaliste > l'équation de rendu [Kajiya 1986]

Deux façons de voir la même chose :



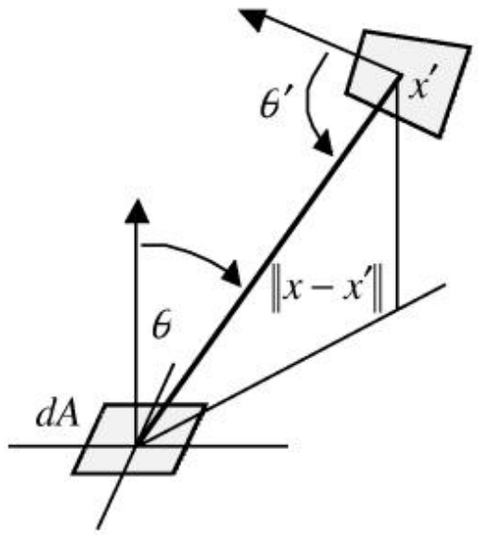
Intégration sur l'hémisphère

occlusion



Intégration sur les surfaces

Rendu réaliste > l'équation de rendu [Kajiya 1986]



$$d\omega = \frac{\cos\theta'}{\|x - x'\|^2} dA'$$

$$\cos\theta \, d\omega = \frac{\cos\theta \cos\theta'}{\|x - x'\|^2} dA'$$

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{M^2} f_r(x, \omega_i(x - x') \rightarrow \omega_o) G(x, x') L_o(x', \omega'_o(x - x')) \, dA'$$

Integrate over All surfaces

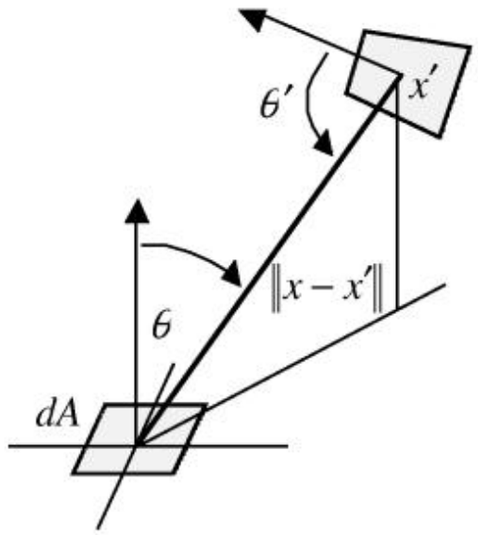
Geometry term

$$G(x, x') = \frac{\cos\theta_i \cos\theta'_o}{\|x - x'\|^2} V(x, x')$$

Visibility term

$$V(x, x') = \begin{cases} 1 & \text{visible} \\ 0 & \text{not visible} \end{cases}$$

Rendu réaliste > l'équation de rendu [Kajiya 1986]



$$d\omega = \frac{\cos\theta'}{\|x - x'\|^2} dA'$$

$$\cos\theta \, d\omega = \frac{\cos\theta \cos\theta'}{\|x - x'\|^2} dA'$$

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{M^2} f_r(x, \omega_i(x - x') \rightarrow \omega_o) G(x, x') L_o(x', \omega'_o(x - x')) \, dA'$$

Integrate over All surfaces

Geometry term

$$G(x, x') = \frac{\cos\theta_i \cos\theta'_o}{\|x - x'\|^2} V(x, x')$$

$$V(x, x') = \begin{cases} 1 & \text{visible} \\ 0 & \text{not visible} \end{cases}$$

Visibility term

Rendu réaliste > résoudre l’équation de rendu

Les algorithmes d’illumination globale

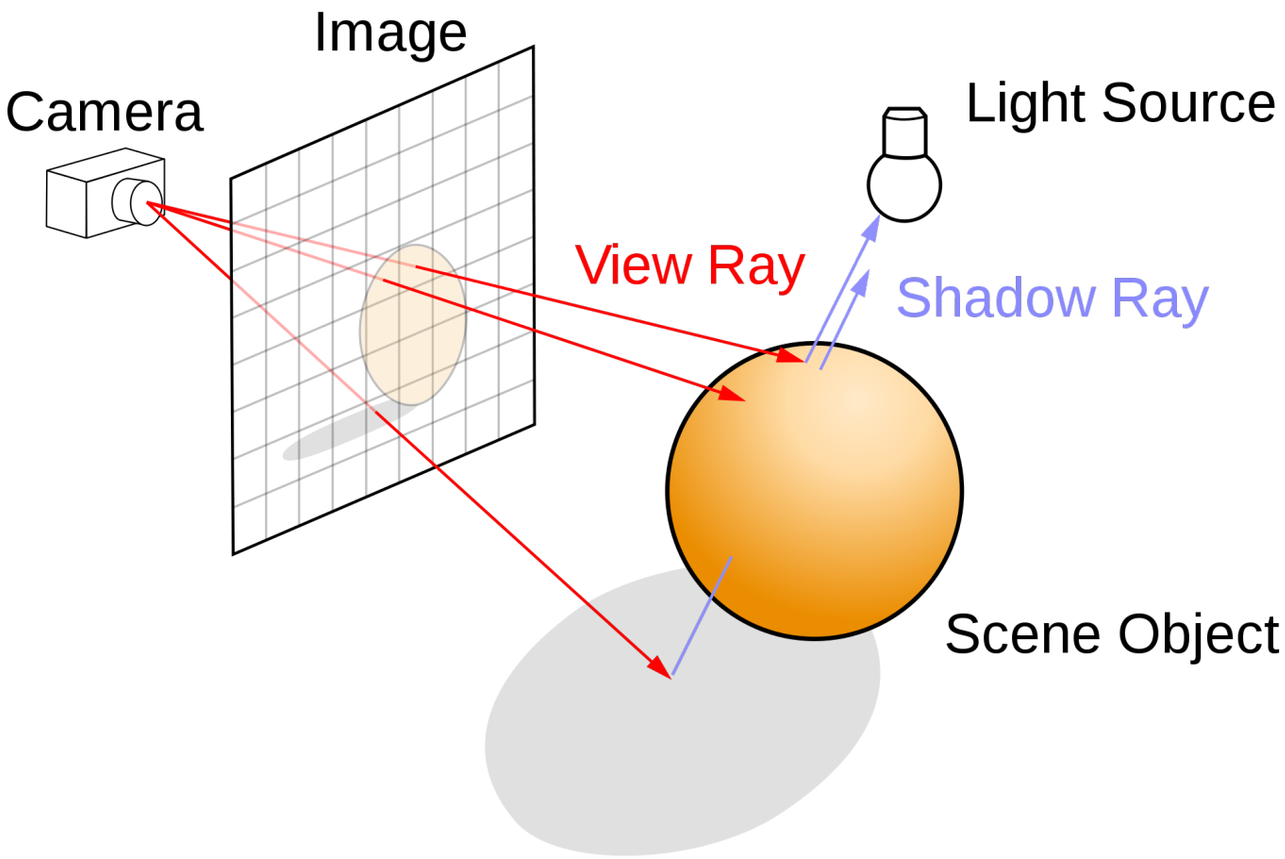
Algorithmes basés éléments finis (subdivision de la géométrie en patchs)

- Exemple : Radiosité

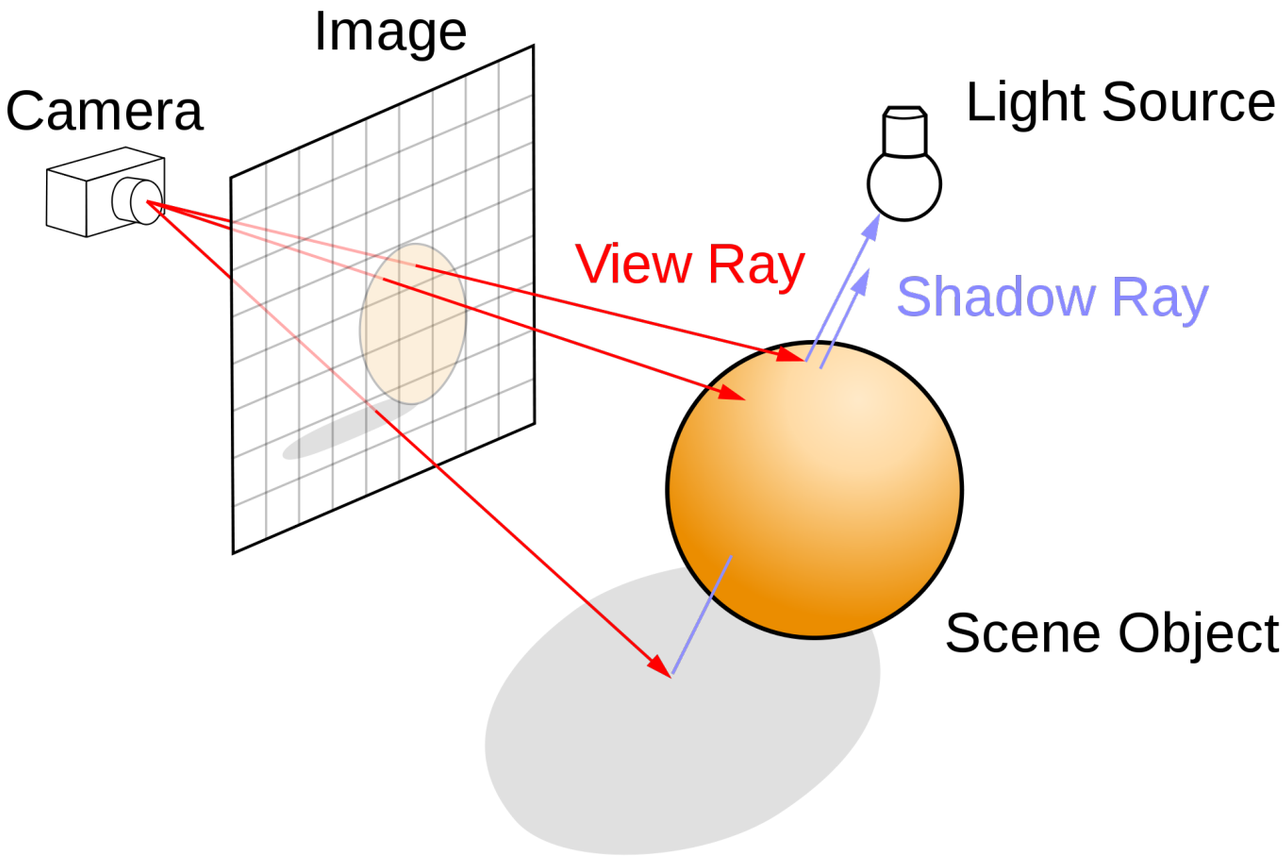
Algorithme basé Monte Carlo, échantillonnage et transformation de l’intégrale en somme de contributions point à point

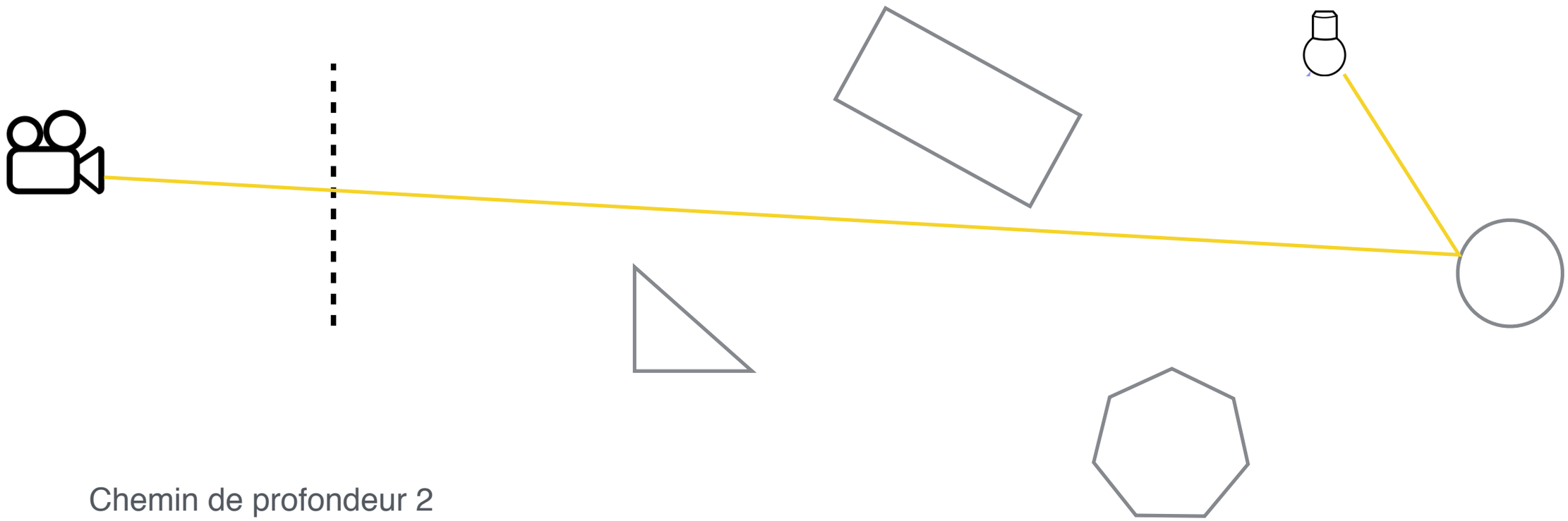
- Examples biaisés : Photon Mapping
- Examples non biaisés : Path Tracing, Bidirectional Path Tracing, Metropolis Light Transport

Rendu réaliste > résoudre l'équation de rendu



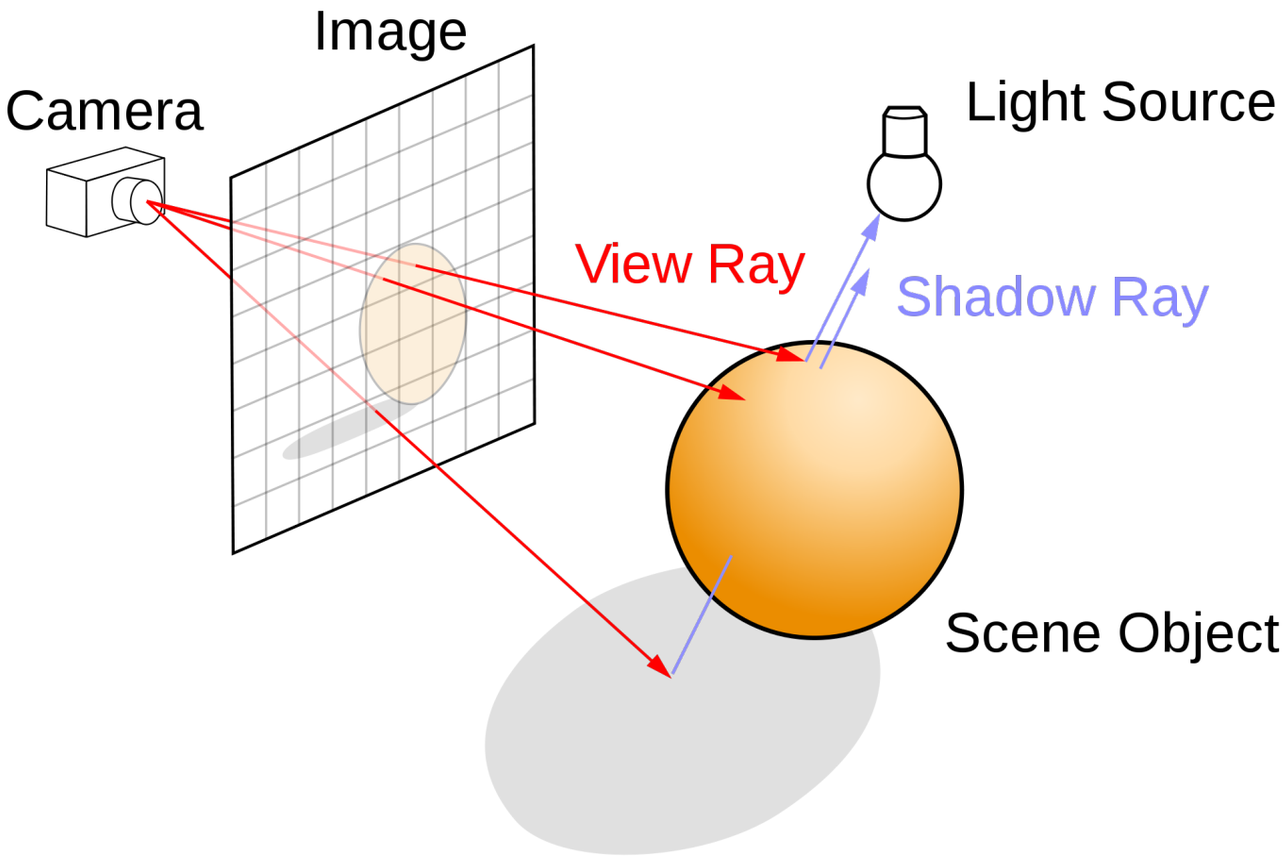
Rendu réaliste > résoudre l'équation de rendu

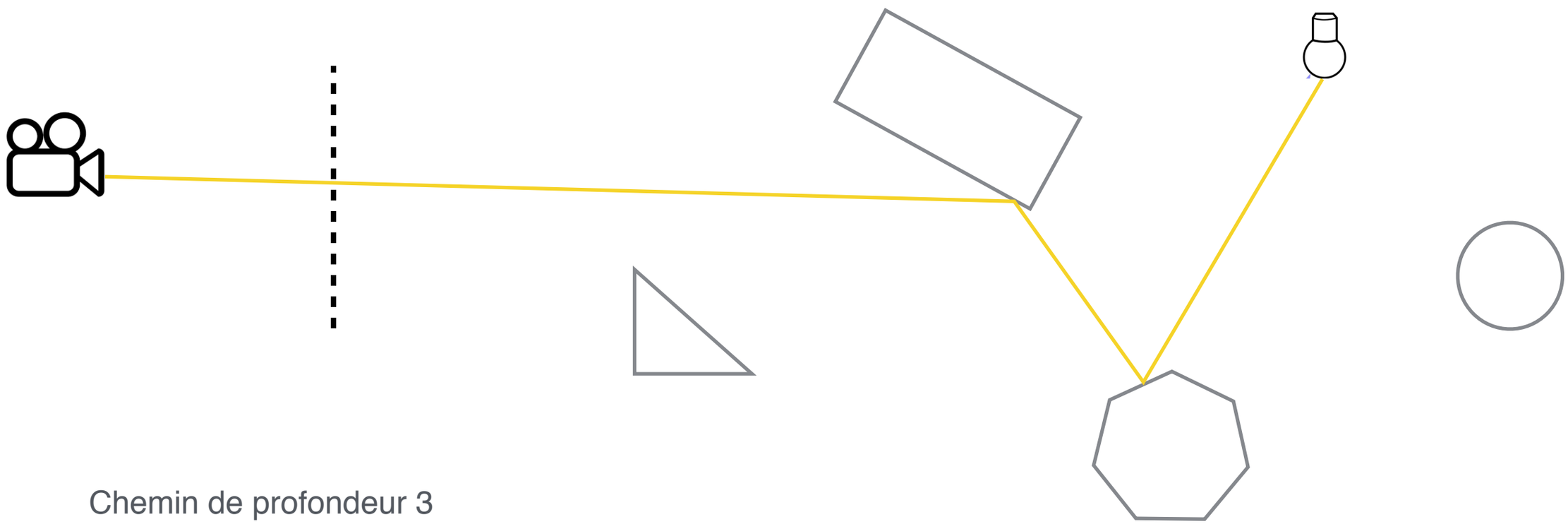




Chemin de profondeur 2

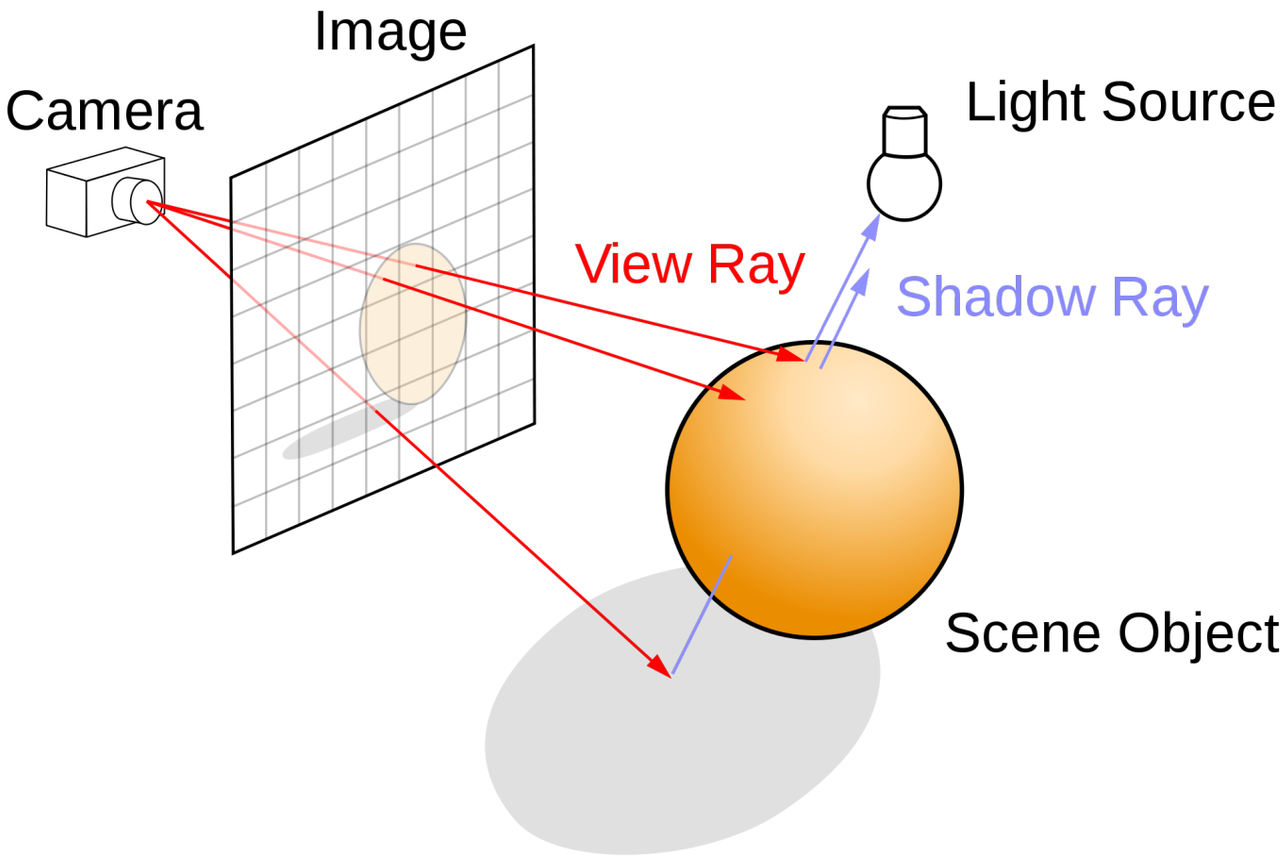
Rendu réaliste > résoudre l'équation de rendu

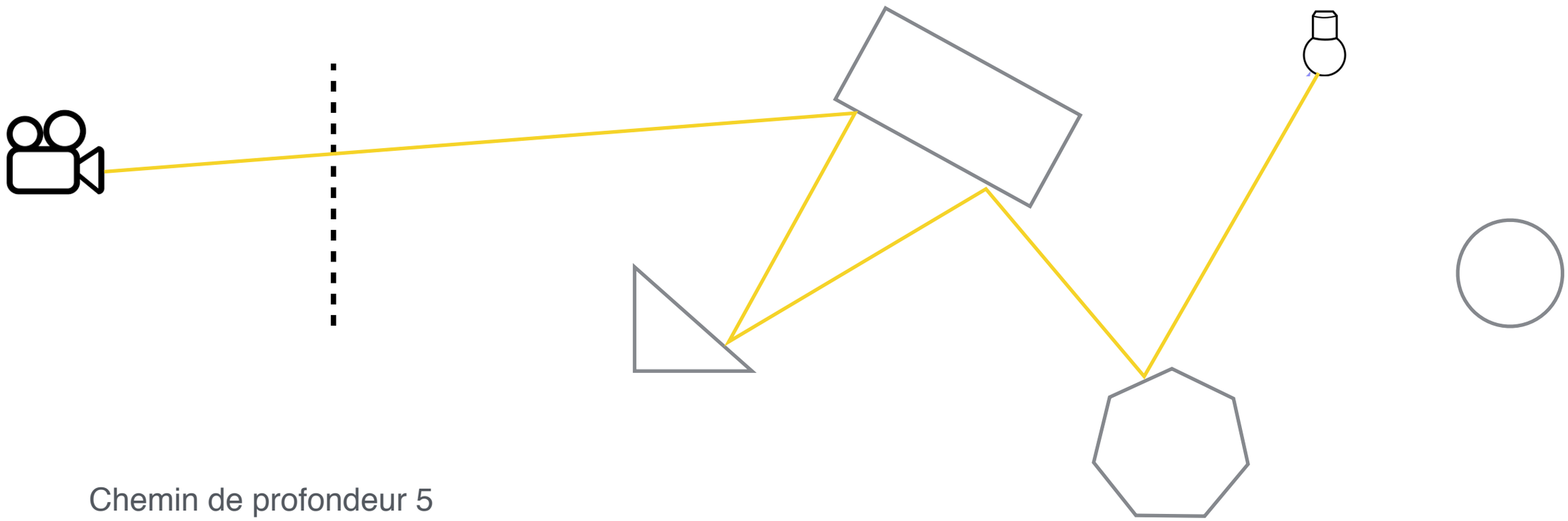




Chemin de profondeur 3

Rendu réaliste > résoudre l'équation de rendu





Chemin de profondeur 5

Rendu réaliste > résoudre l’équation de rendu

- Il faut beaucoup d’échantillons (de chemins) pour que la somme converge vers l’intégrale

- Certain phénomène (réflections spéculaires, caustiques…) sont plus difficiles à échantillonner.

- Beaucoup de travaux d’un point de vue statistique et probabiliste pour améliorer les stratégies d’échantillonnage et donc la convergence du résultat

- Un sous-échantillonnage engendre du bruit dans l’image

Démo (Mitsuba)

Rendu réaliste > tout vient à point à qui sait attendre (longtemps)

$$L_o(x,\omega_o) = L_e(x,\omega_o) + \int_{M^2} f_r(x,\omega_i(x-x') \to \omega_o) G(x,x') L_o(x',\omega'_o(x-x'))\, dA'$$

Integrate over All surfaces

Geometry term

$$G(x,x') = \frac{\cos\theta_i \cos\theta'_o}{\|x-x'\|^2} V(x,x')$$

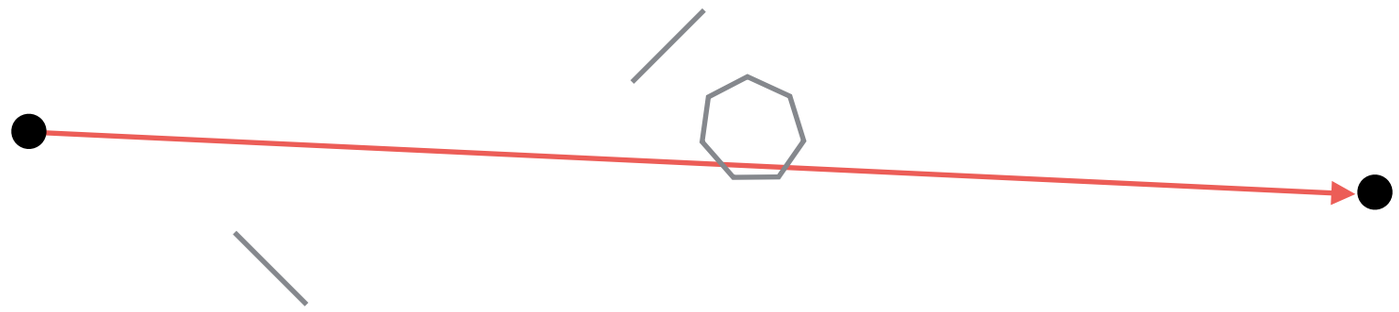
$$V(x,x') = \begin{cases} 1 & \text{visible} \\ 0 & \text{not visible} \end{cases}$$

Visibility term

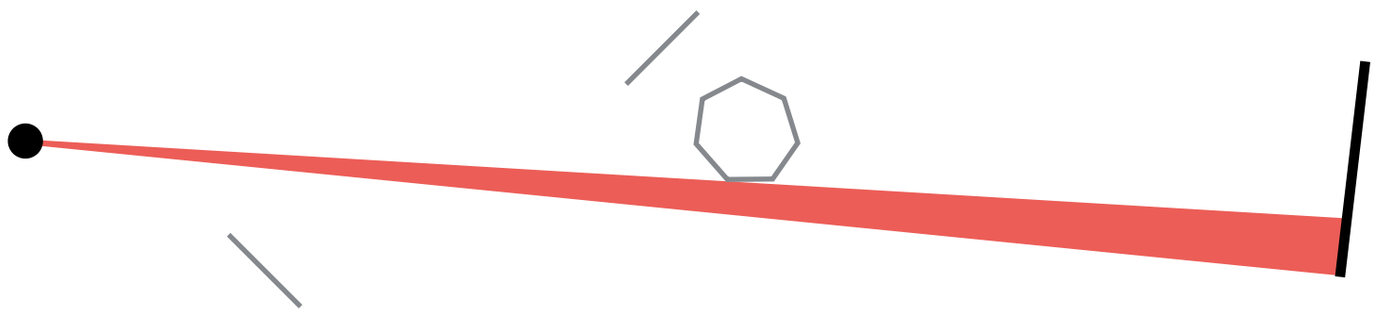
- plus de 90% des temps de calculs sont imputables au terme V…
- le Graal : faire du rendu réaliste en temps réel (0.016s par image…) Brigade (youtube)

Visibilité > un enjeu, plusieurs niveaux de complexité

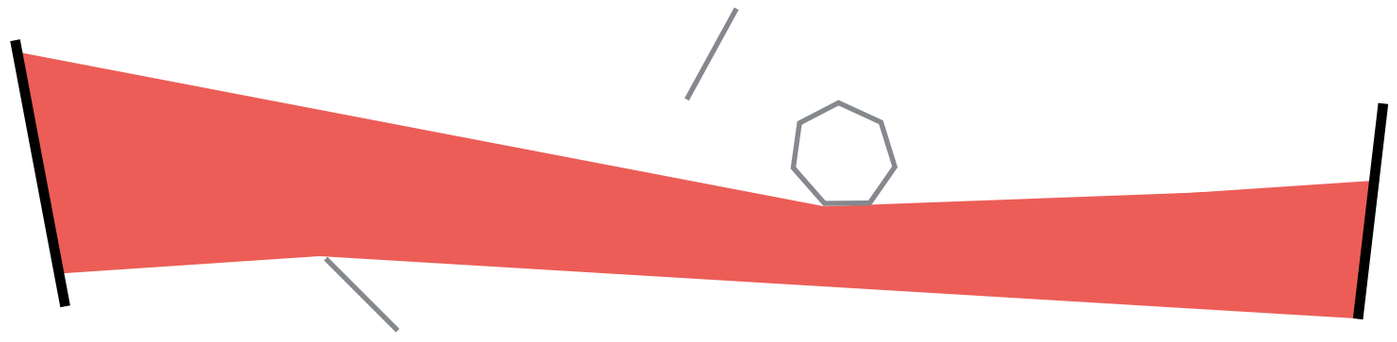
- visibilité point à point (le long d'un segment de droite)



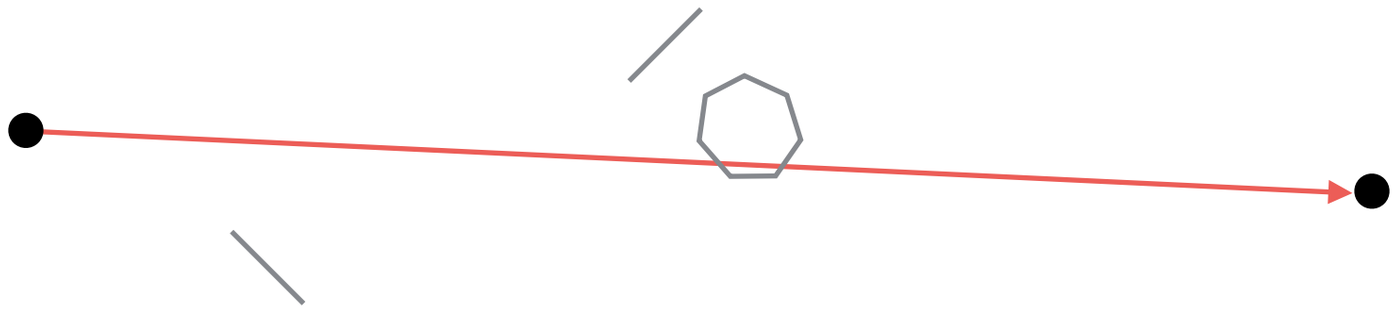
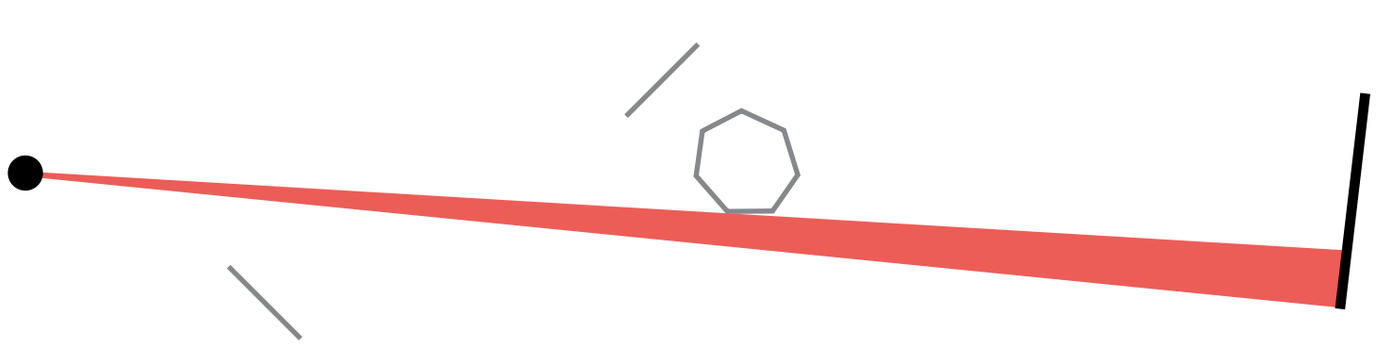
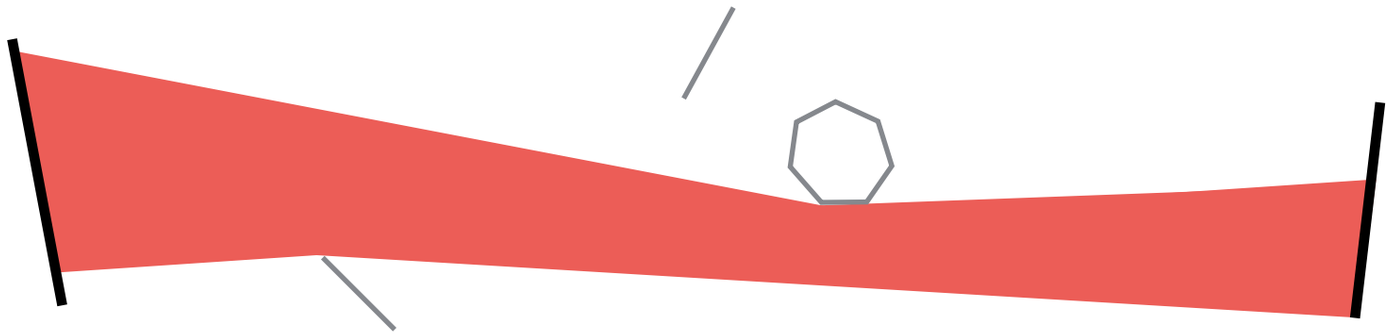
- visibilité point à segment/polygone (ou depuis un point)



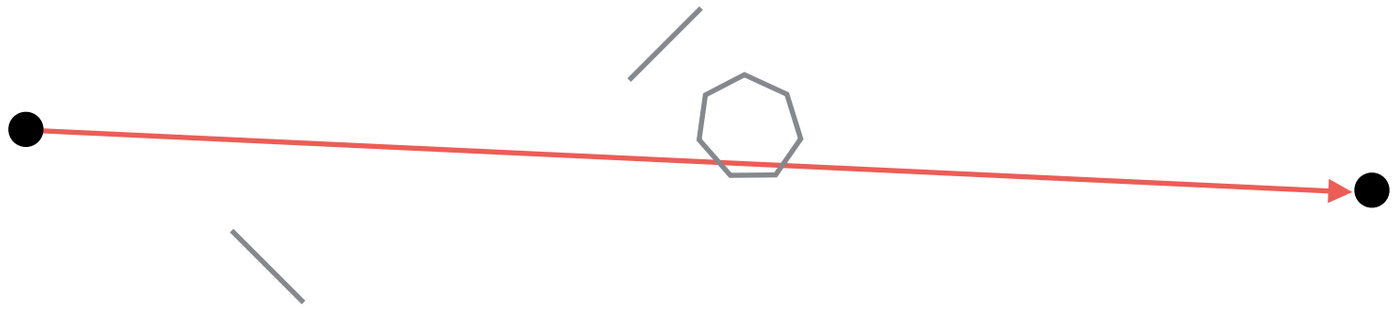
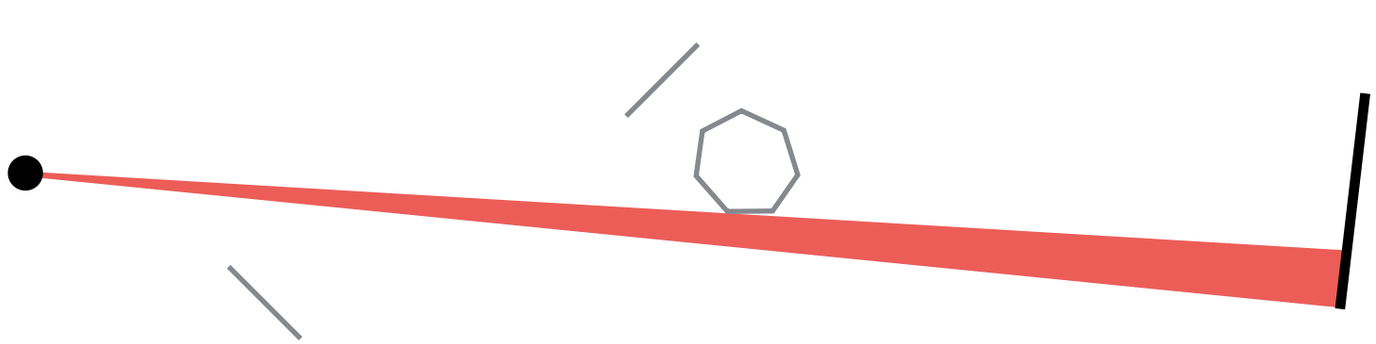
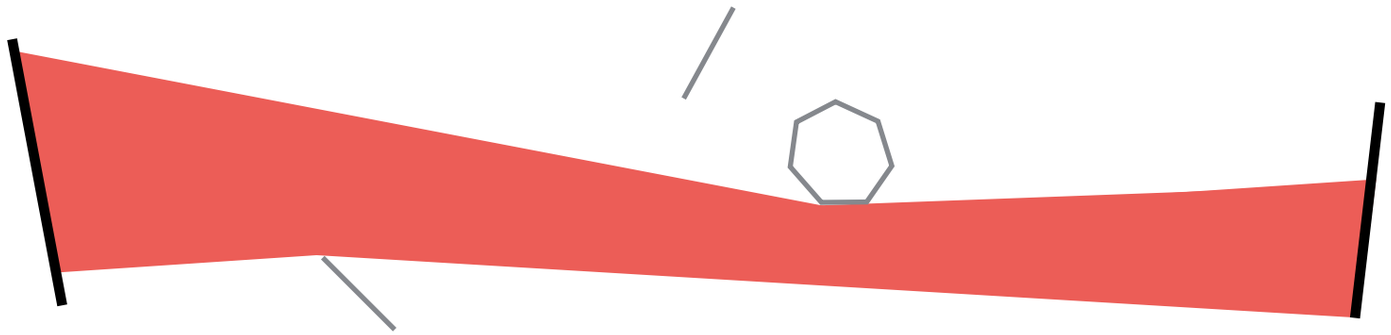
- visibilité segment à segment ou polygone à polygone



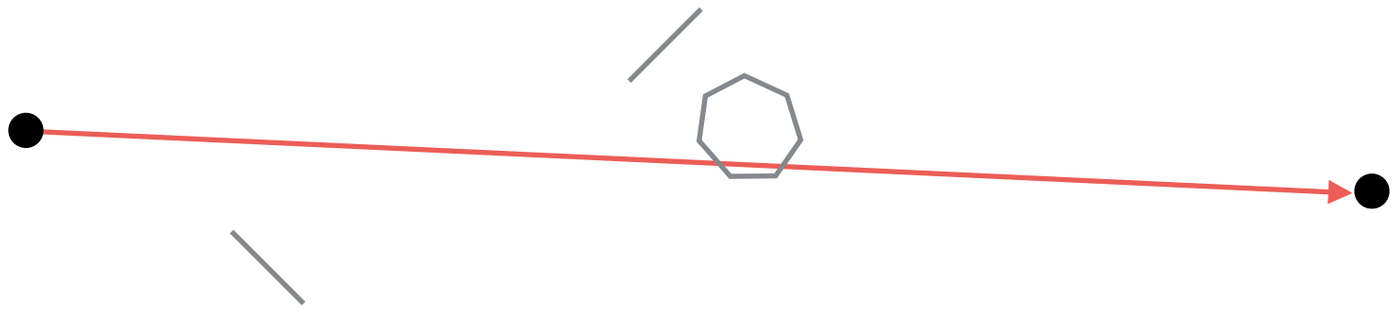
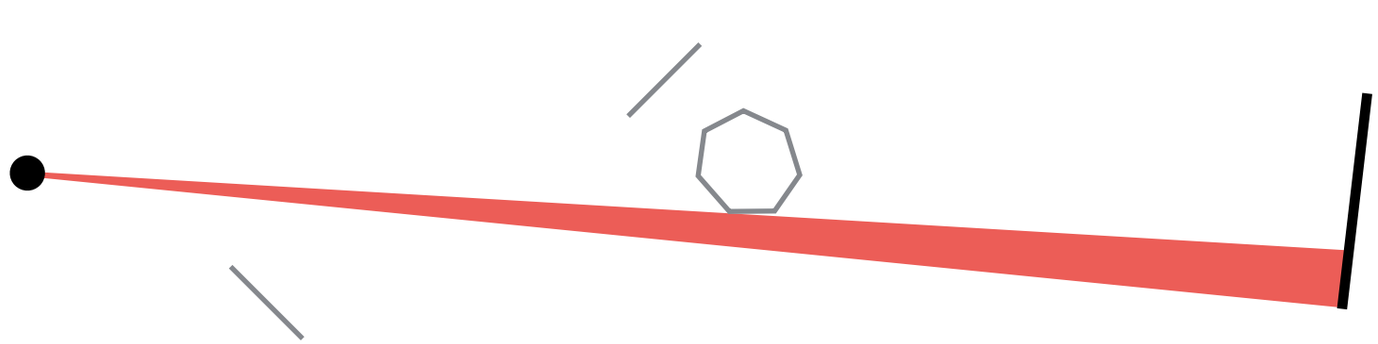
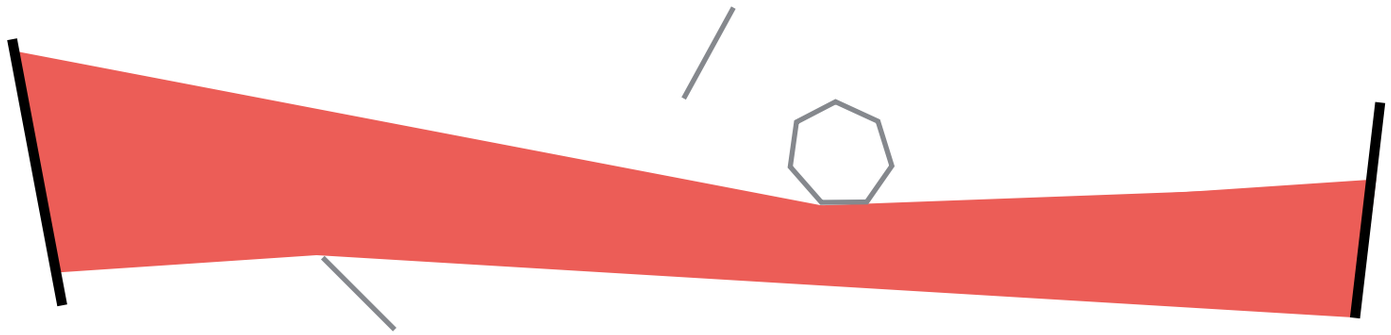
Visibilité > un enjeu, plusieurs niveaux de complexité

	2D	3D
• visibilité point à point (le long d'un segment de droite) 		
• visibilité point à segment/polygone (ou depuis un point) 		
• visibilité segment à segment ou polygone à polygone 		

Visibilité > un enjeu, plusieurs niveaux de complexité

	2D	3D
• visibilité point à point (le long d'un segment de droite) 	0D	0D
• visibilité point à segment/polygone (ou depuis un point) 	1D	2D
• visibilité segment à segment ou polygone à polygone 	2D	4D

Visibilité > un enjeu, plusieurs niveaux de complexité

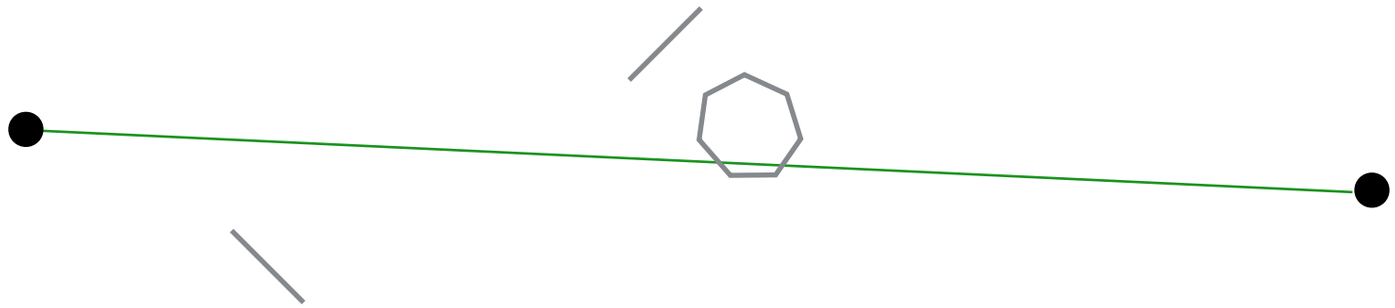
	2D	3D
• visibilité point à point (le long d'un segment de droite) 	0D	0D
• visibilité point à segment/polygone (ou depuis un point) 	1D	2D
• visibilité segment à segment ou polygone à polygone 	2D	4D

Visibilité > un enjeu, plusieurs niveaux de complexité

- La complexité d’un problème de visibilité peut s’apprécier à la dimension de l’ensemble de droites sous-jacent
- En 3D, il existe un saut de complexité lorsqu’on passe de la visibilité depuis un point à la visibilité depuis une surface
- Appréhender un problème 4D par nature… pas forcément intuitif

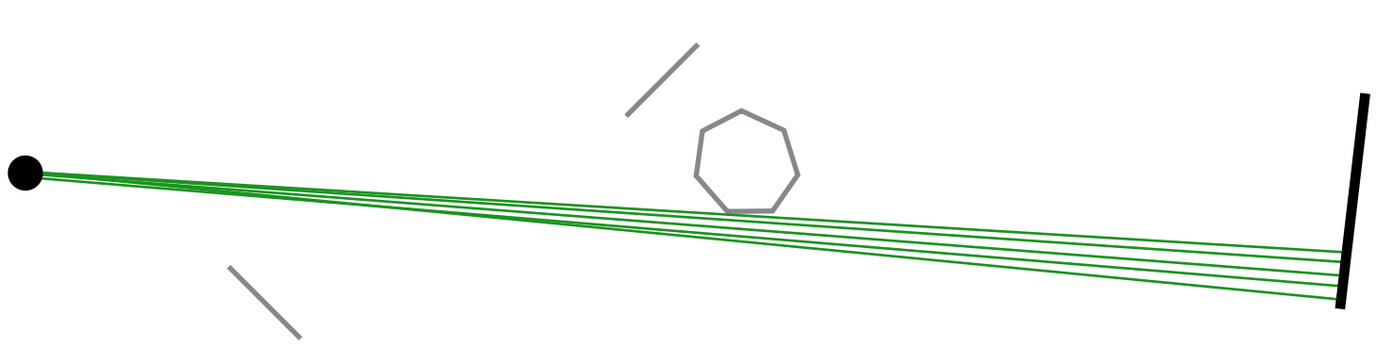
Visibilité > solution (?) à tout faire : sampling

- visibilité point à point (le long d'un segment de droite)



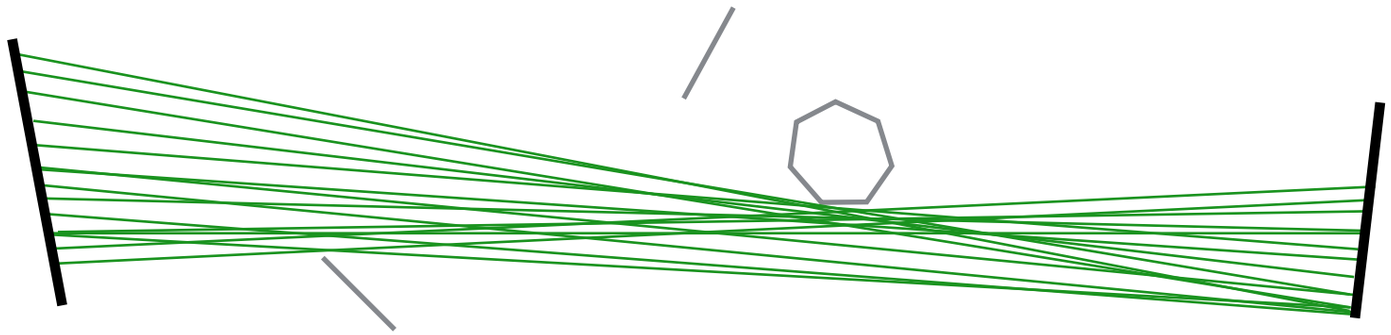
0D

- visibilité point à polygone (ou depuis un point)



2D

- visibilité polygone à polygone



4D

Visibilité > solution (?) à tout faire : sampling

- Principe de base : linéariser un problème complexe
  - ici échantillonner un ensemble 4D en n problèmes 0D

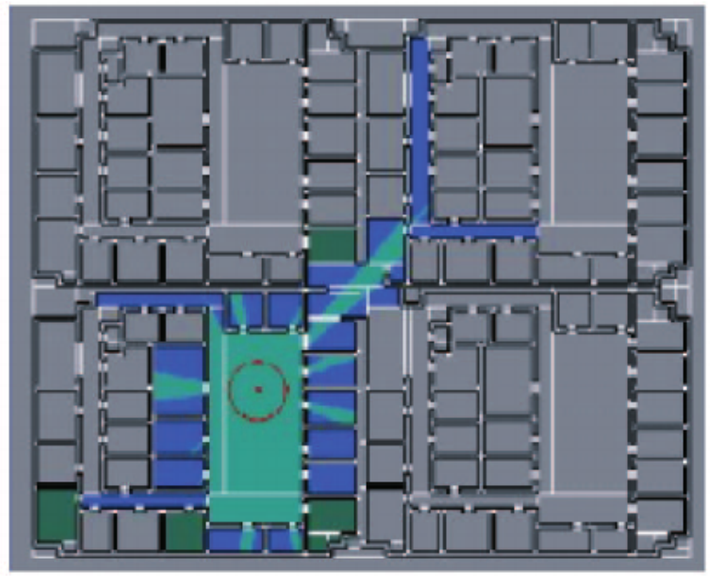
- Mais, n’est pas forcément synonyme de rapidité
  - cf, la démo Mitsuba et les méthodes de rendu basées Monte Carlo

- Mais, introduit la problématique du sous-échantillonnage (bruit)

- N’est pas une solution à tout faire, ne peut résoudre certains problèmes
  - échantillonner à coups de rayons/droites, c’est explorer une pièce noire avec un pointeur laser

Visibilité > limites du sampling

- Example : prouver que deux polygones sont visibles, ou invisibles
  - combien d’échantillons par garantir une réponse ? Une infinité…
  - Application pratique : Potentially Visible Sets

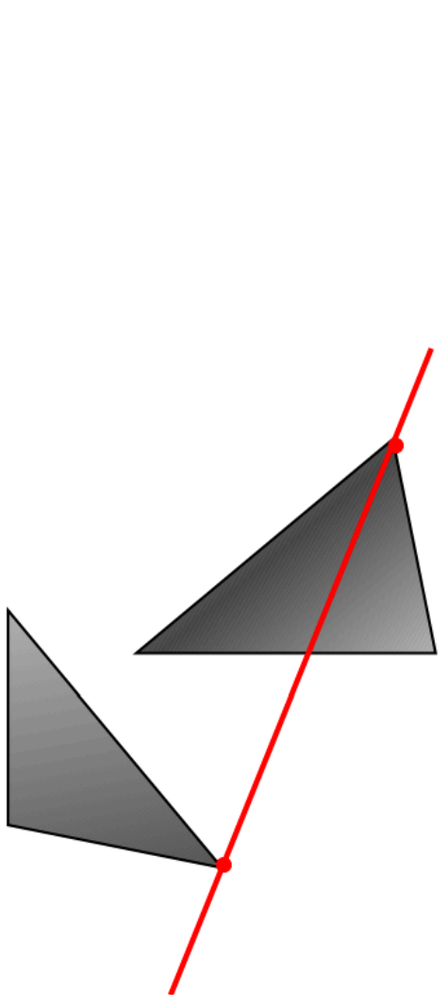


- Example : est-ce que la vue change au voisinage d’un point ?
  - comprendre, si l’on se déplace d’un epsilon (ou plus), est-ce que des objets apparaissent ou disparaissent ?
  - ou plus simplement, si je perturbe un rayon d’un epsilon, est-ce que sa visibilité change ?
  - introduit la notion de cohérence visuelle

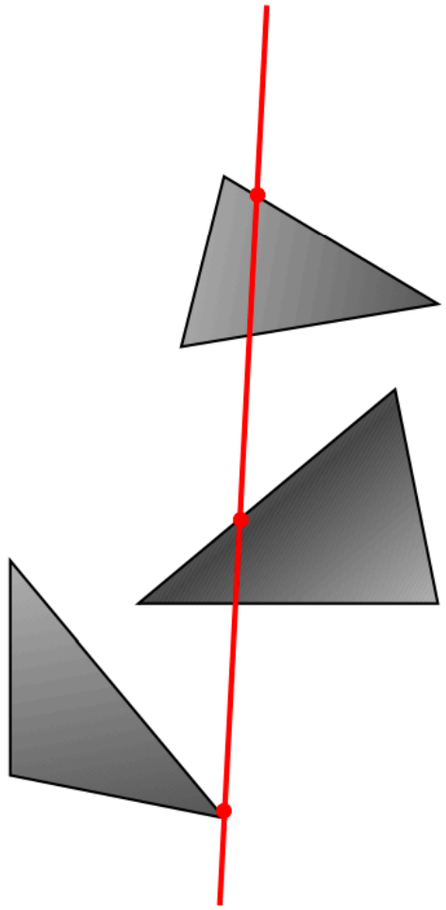
Besoin d’une caractérisation analytique de la visibilité

Visibilité > visibilité analytique, étude en 3D

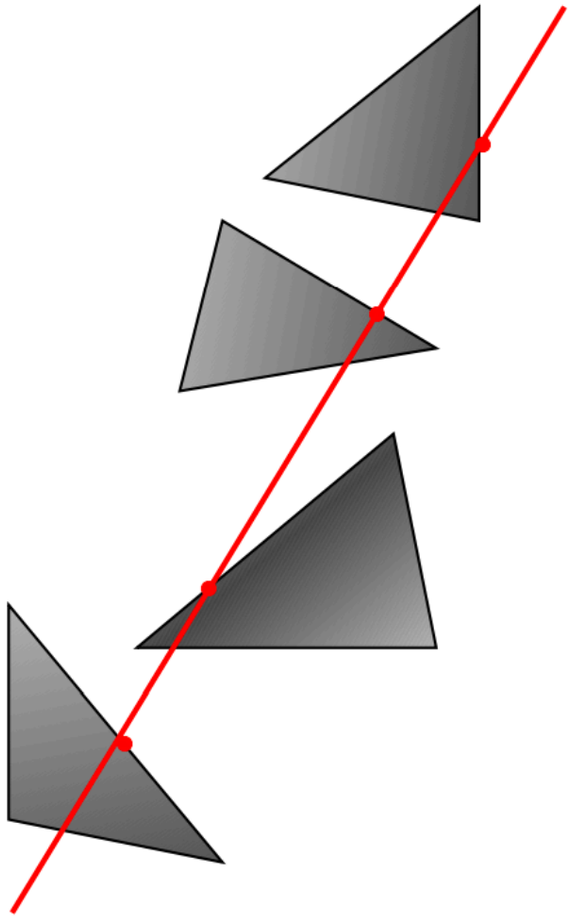
- [Durand99], étude des discontinuités dans visibilité



(vv)



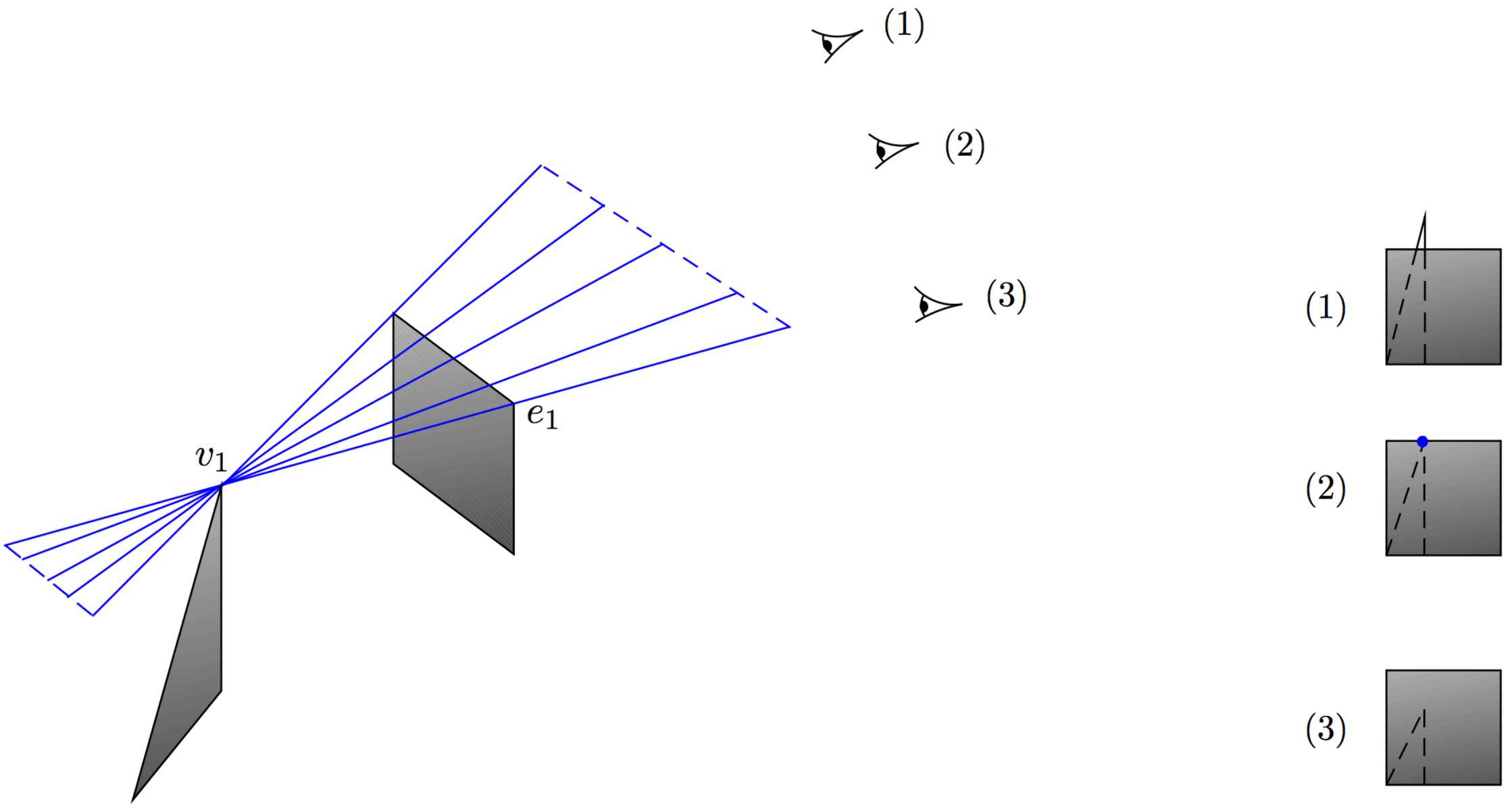
(vee)



(eeee)

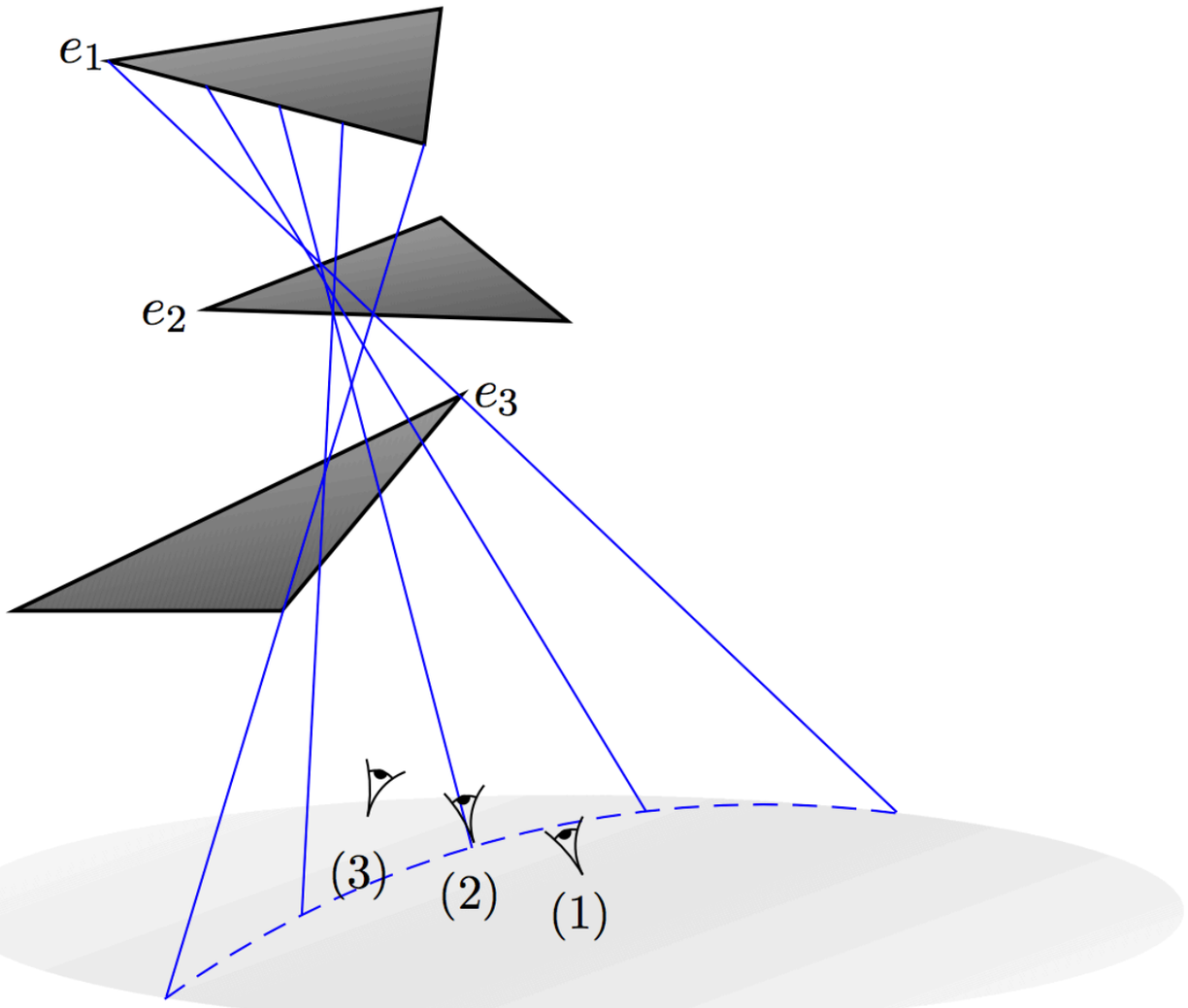
Visibilité > visibilité analytique, étude en 3D

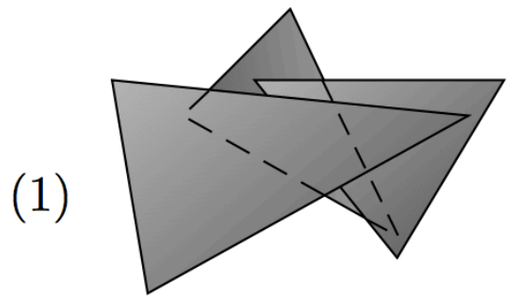
- [Durand99], étude des discontinuités dans visibilité (ev)

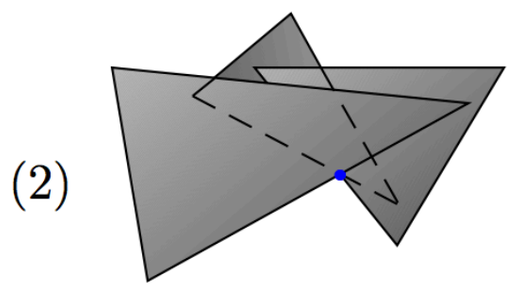


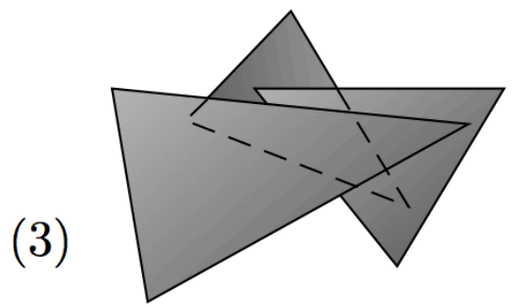
Visibilité > visibilité analytique, étude en 3D

- [Durand99], étude des discontinuités dans visibilité (eee)



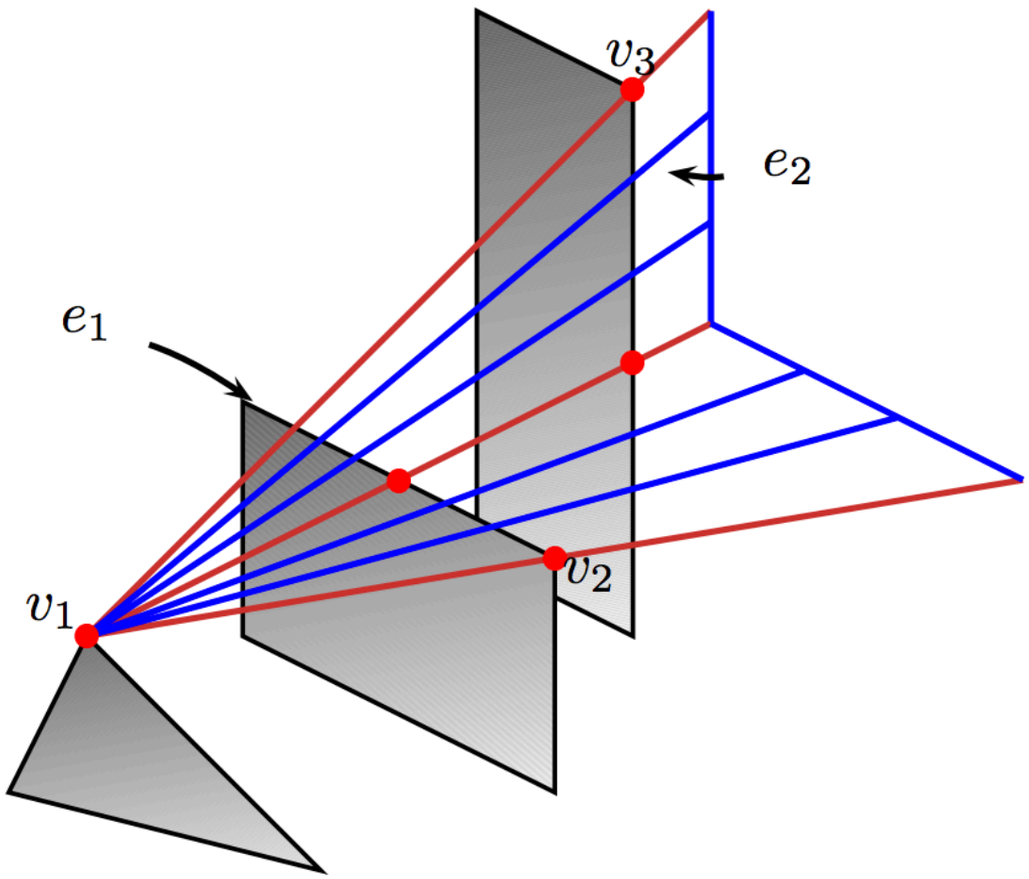


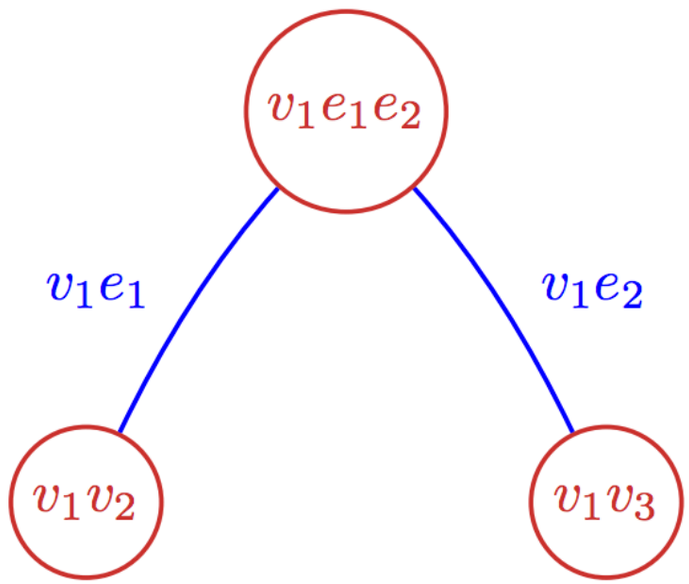




Visibilité > visibilité analytique, étude en 3D

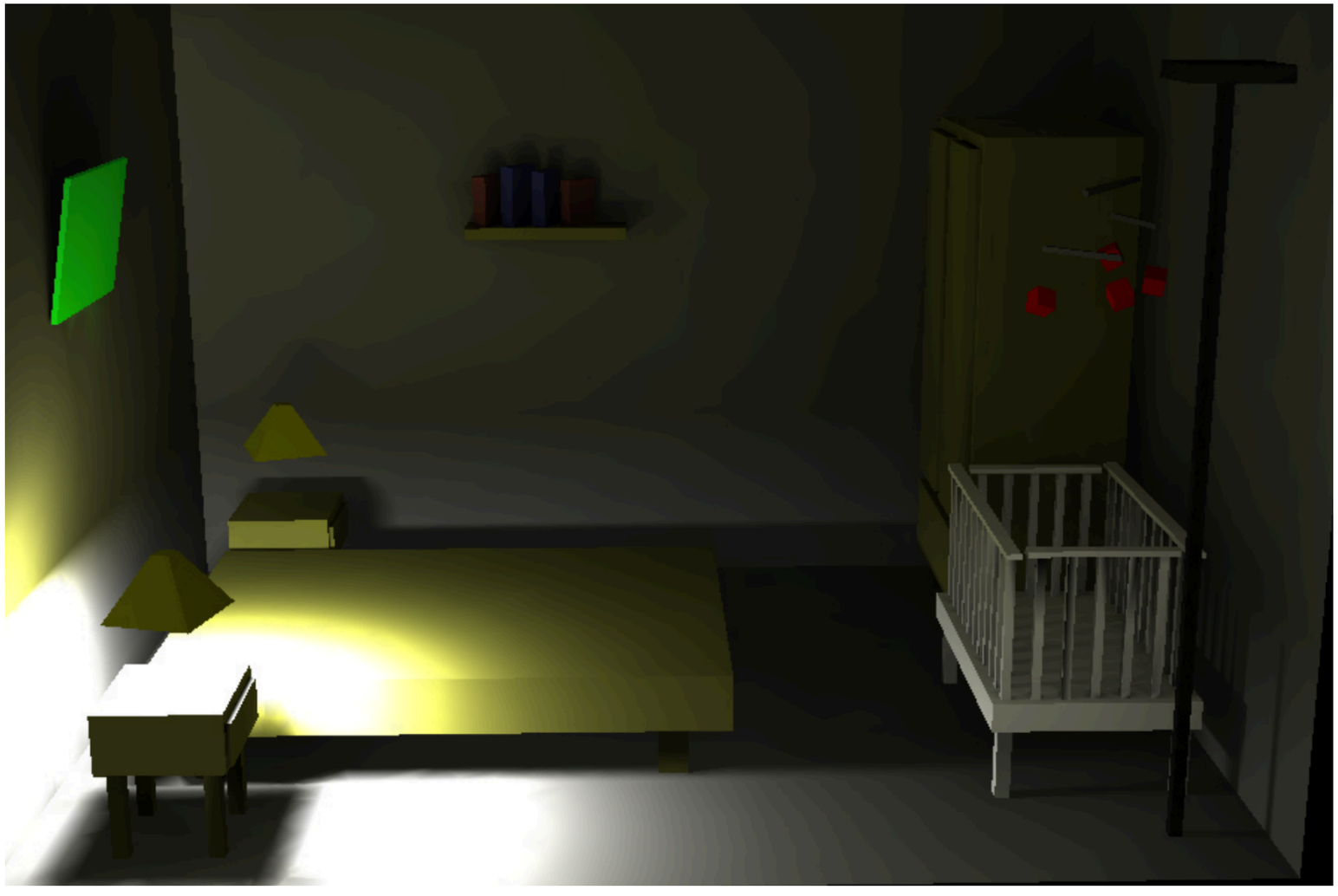
- [Durand99], construction d'un graphe dont les noeuds sont des événements visuels

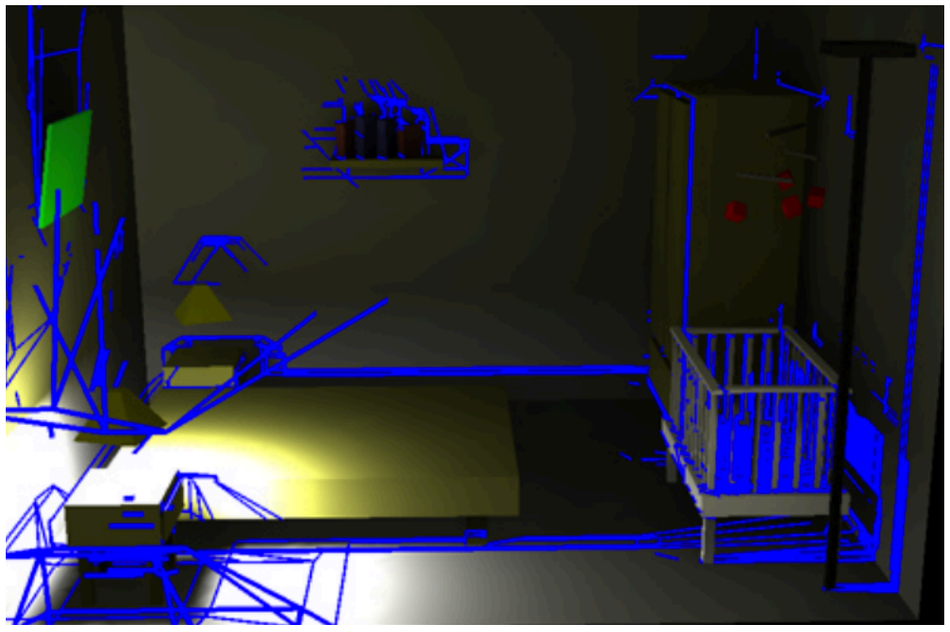


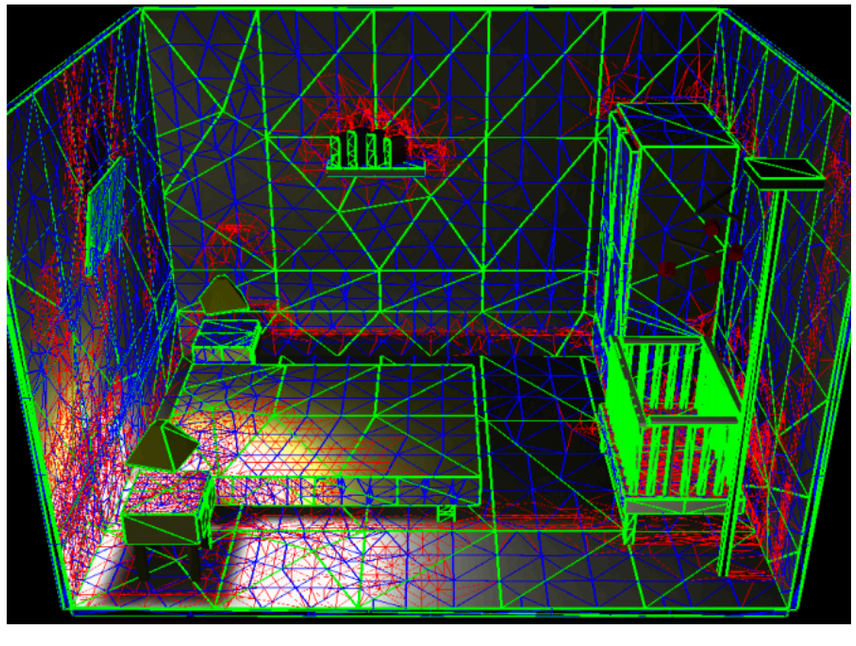


Visibilité > visibilité analytique, étude en 3D

- [Durand99], application en radiosité (discontinuity meshing)

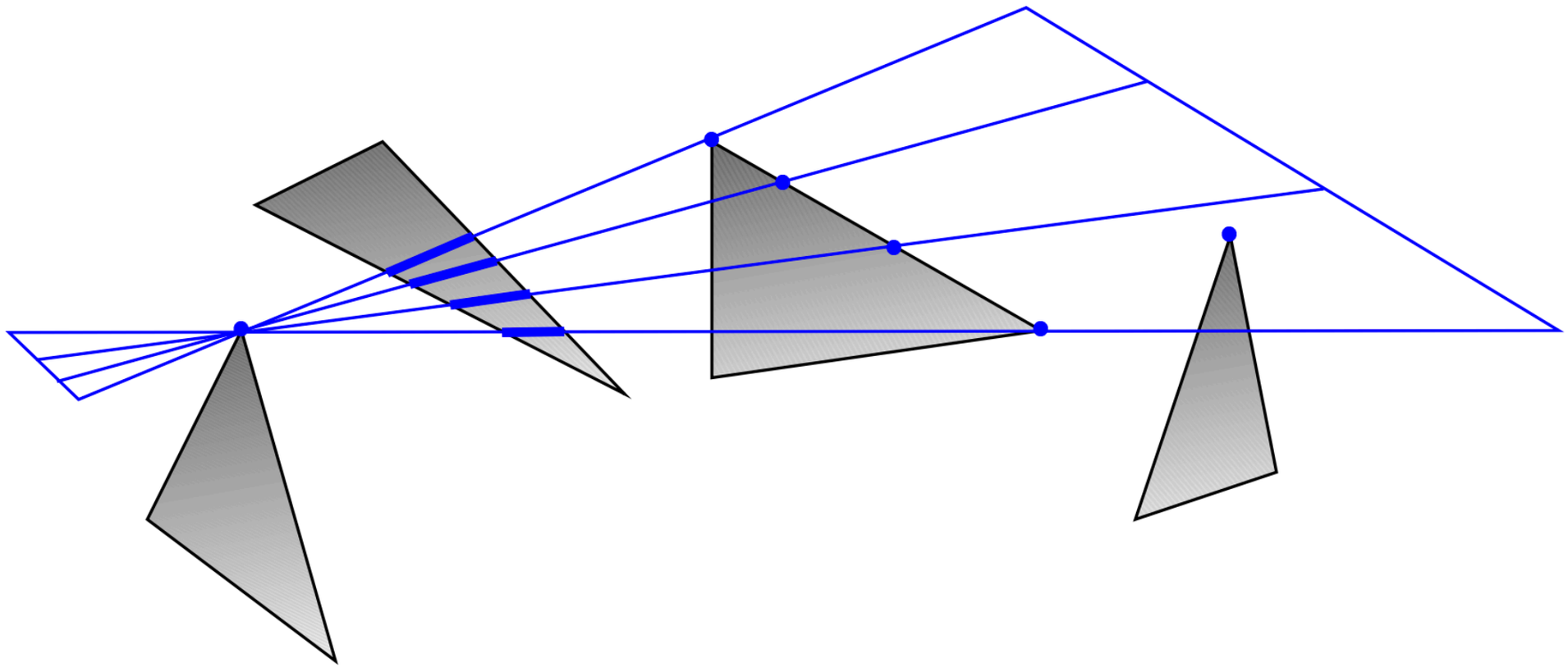






Visibilité > visibilité analytique, étude en 3D

- [Durand99], impossible de gérer tous les cas « dégénérés »



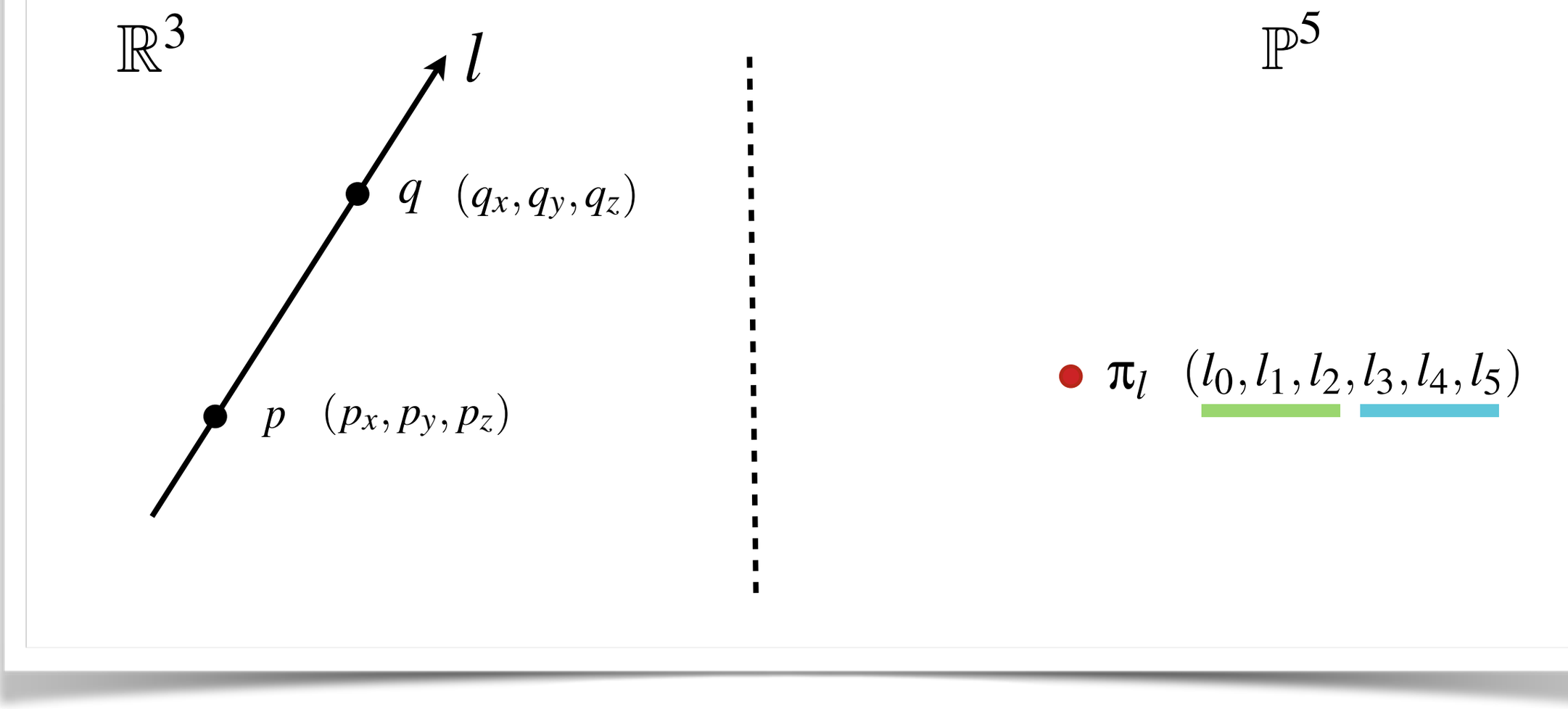
- Il faut monter en dimension, aller dans un espace au moins égale à la dimension du problème

Espace de Plucker > Julius Plucker

Julius Plücker (16 juin ou 16 juillet 1801 à Elberfeld, Saint-Empire romain germanique - 22 mai 1868 à Bonn, Prusse) est un mathématicien et un physicien allemand. Il a obtenu des résultats fondamentaux en géométrie analytique et fut un pionnier dans les recherches sur les rayons cathodiques qui aboutirent à la découverte de l'électron. Il a aussi beaucoup travaillé sur les courbes de Lamé.

WIKIPEDIA
The Free Encyclopedia

Espace de Plucker > paramétrisation des droites de $\mathbb{R}^3$



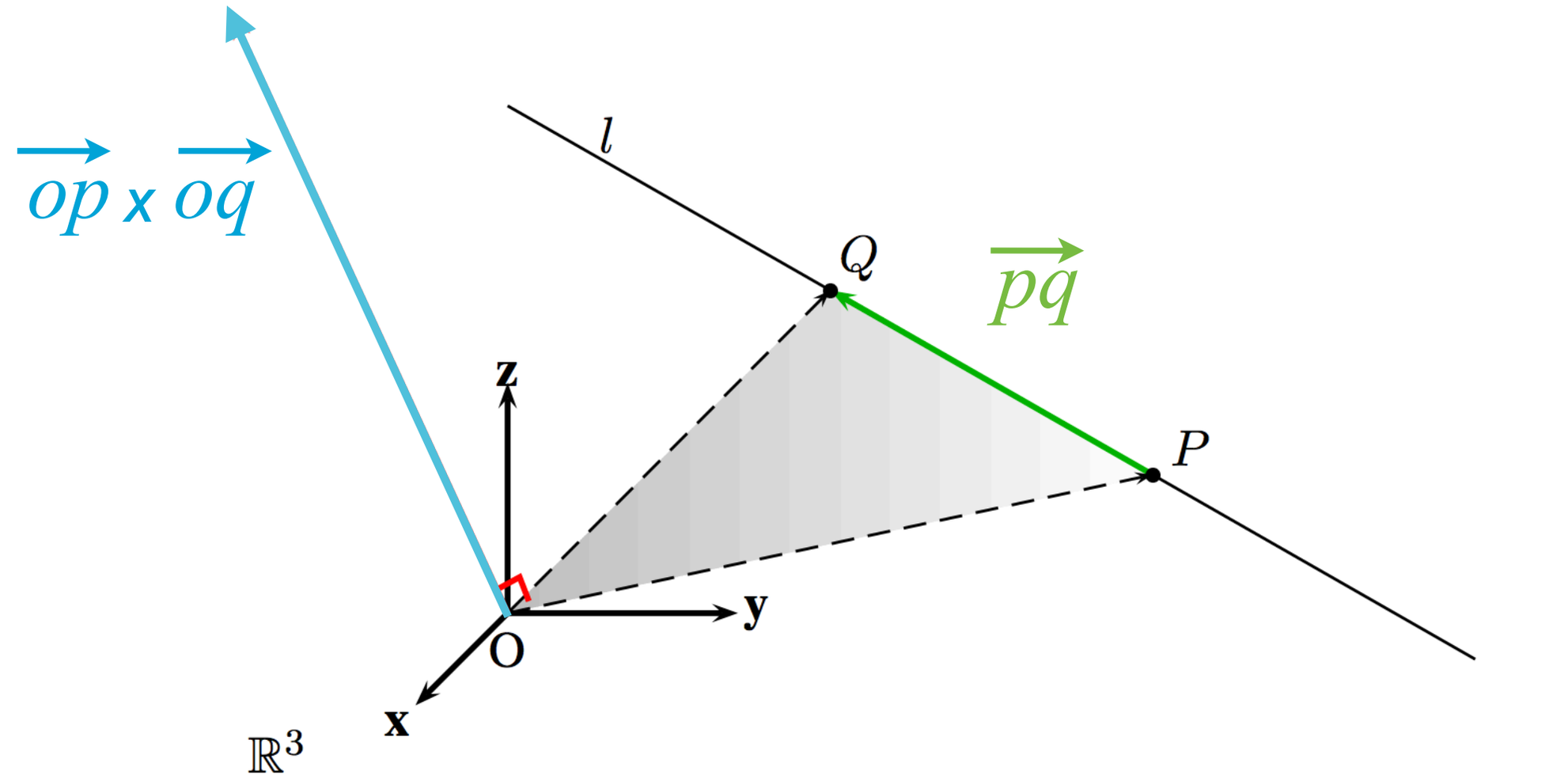
$\overrightarrow{pq}$

$$l_0 = q_x - p_x$$
$$l_1 = q_y - p_y$$
$$l_2 = q_z - p_z$$

$$l_3 = q_z p_y - q_y p_z$$
$$l_4 = q_x p_z - q_z p_x$$
$$l_5 = q_y p_x - q_x p_y$$

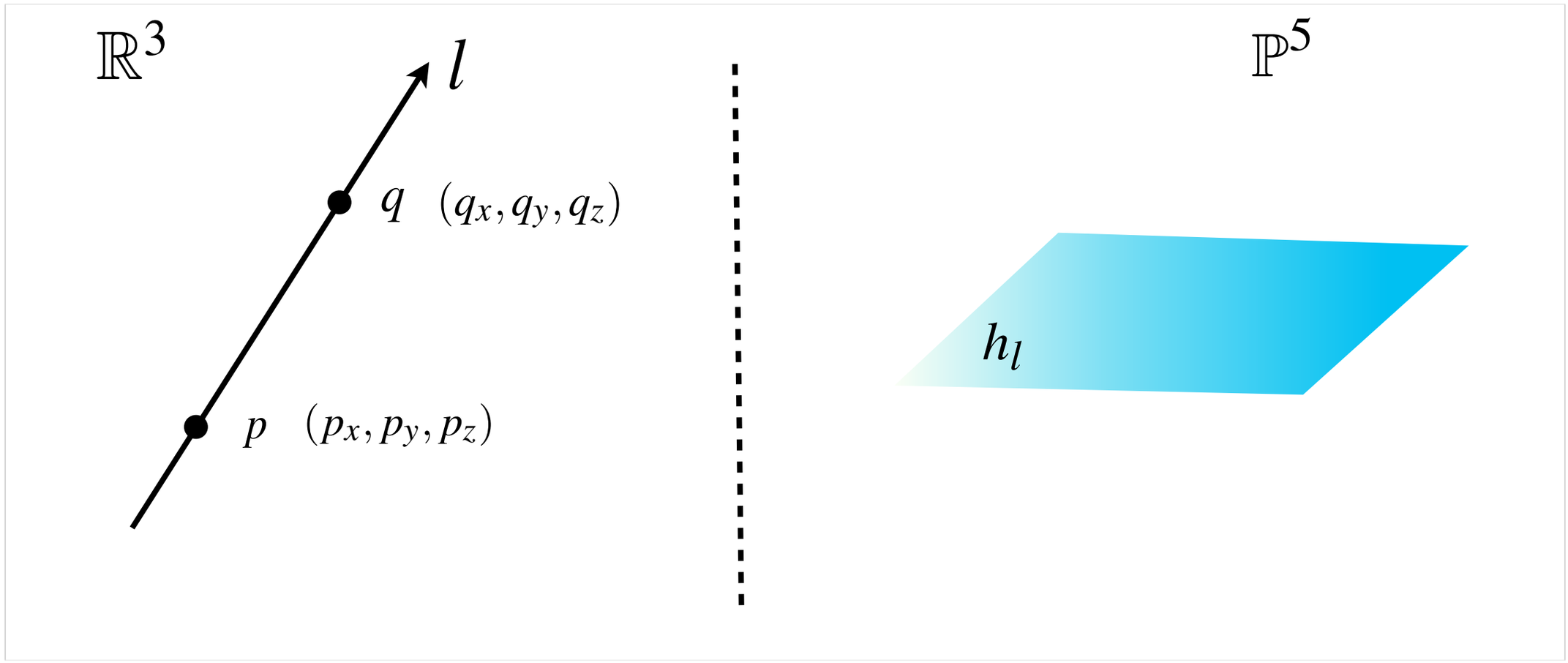
$\overrightarrow{op} \times \overrightarrow{oq}$

Espace de Plucker > paramétrisation des droites de $\mathbb{R}^3$



$$\overrightarrow{pq} \quad \begin{vmatrix} l_0 = q_x - p_x \\ l_1 = q_y - p_y \\ l_2 = q_z - p_z \end{vmatrix} \quad \begin{vmatrix} l_3 = q_z p_y - q_y p_z \\ l_4 = q_x p_z - q_z p_x \\ l_5 = q_y p_x - q_x p_y \end{vmatrix} \quad \overrightarrow{op} \times \overrightarrow{oq}$$

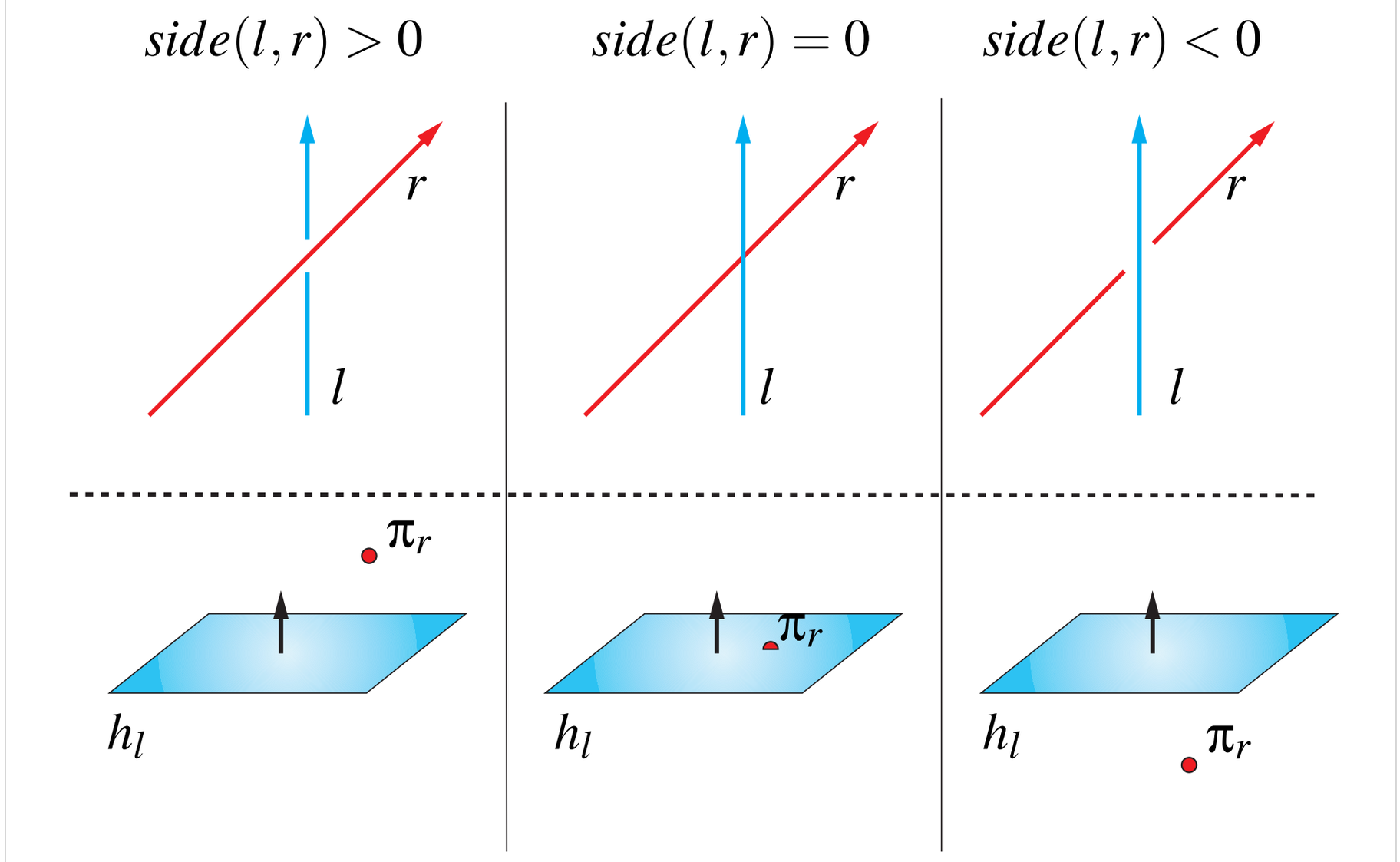
Espace de Plucker > dualité point - hyperplan



- $\pi_l \quad (l_0, l_1, l_2, l_3, l_4, l_5)$

$$h_l(x) = l_3x_0 + l_4x_1 + l_5x_2 + l_0x_3 + l_1x_4 + l_2x_5 = 0$$

Espace de Plucker > orientation relative de 2 droites



$$side(l,r) = l_3r_0 + l_4r_1 + l_5r_2 + l_0r_3 + l_1r_4 + l_2r_5$$

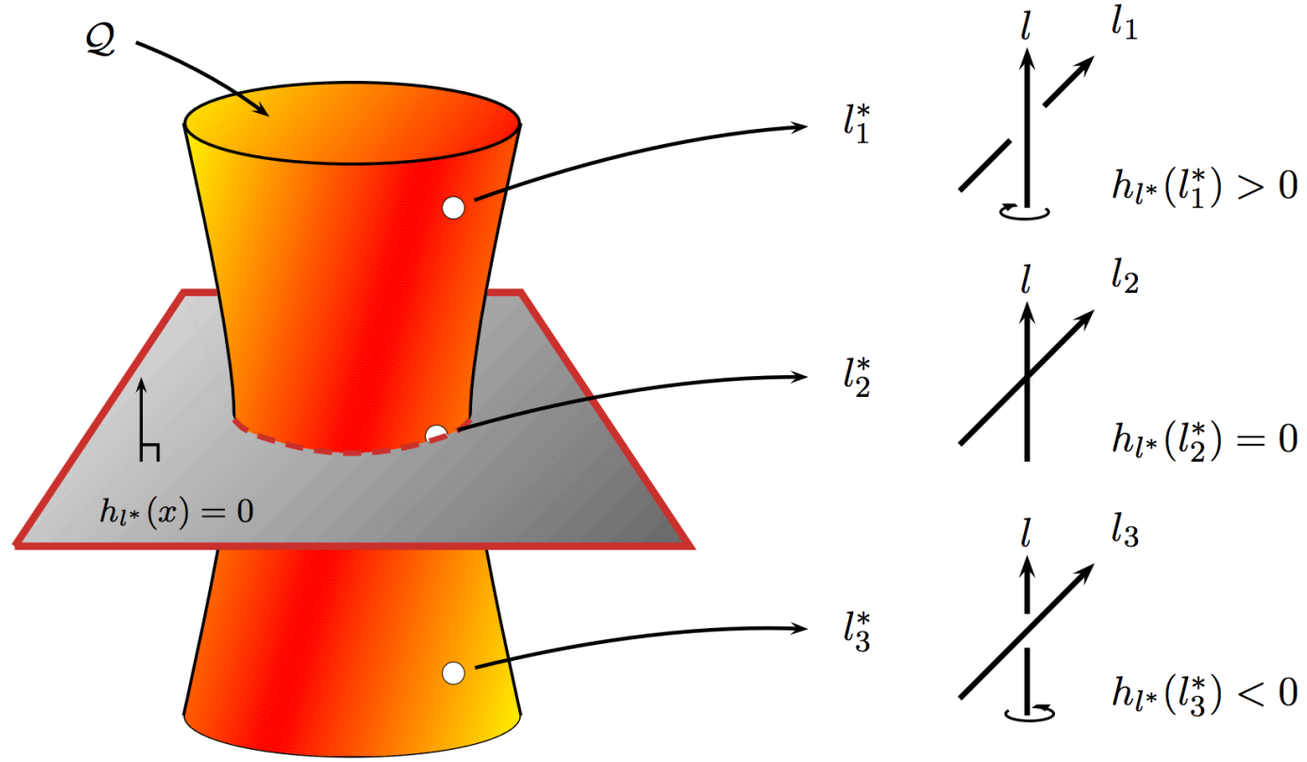
$$= h_l(\pi_r)$$

Espace de Plucker > Quadrique de Plucker

- la paramétrisation des droites réelles est injective

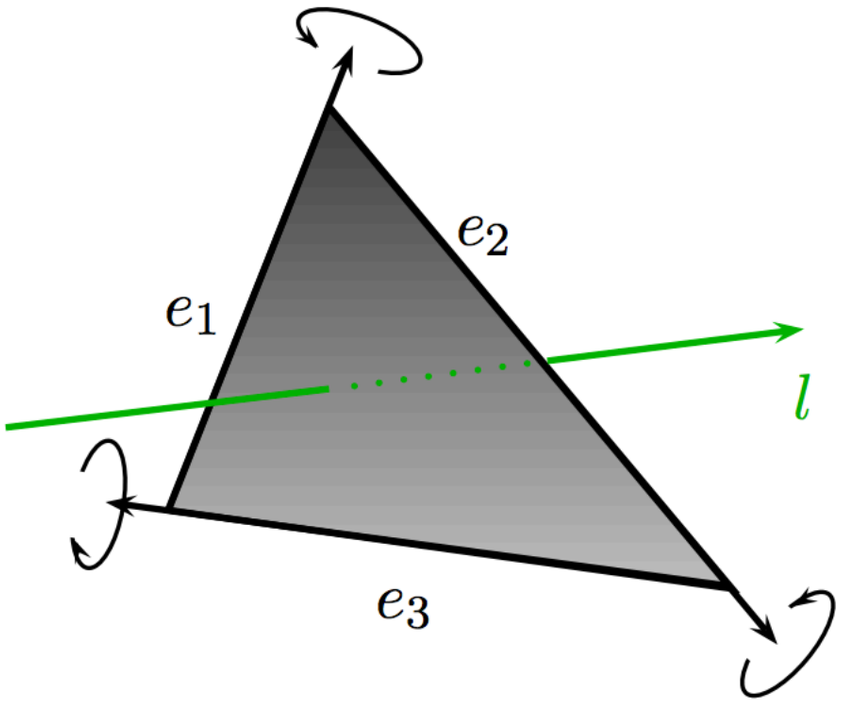
- Dans l'espace de Plucker, l'ensemble des droites réelles forme une quadrique de dimension 4 appelée *hypersurface de Plucker* ou parfois *variété de Grassmann*. Elle est caractérisable comme suit :

$$\mathcal{Q} = \{x \in \mathbb{P}^5 \mid h_x(x) = 0\} \backslash \{0\}$$



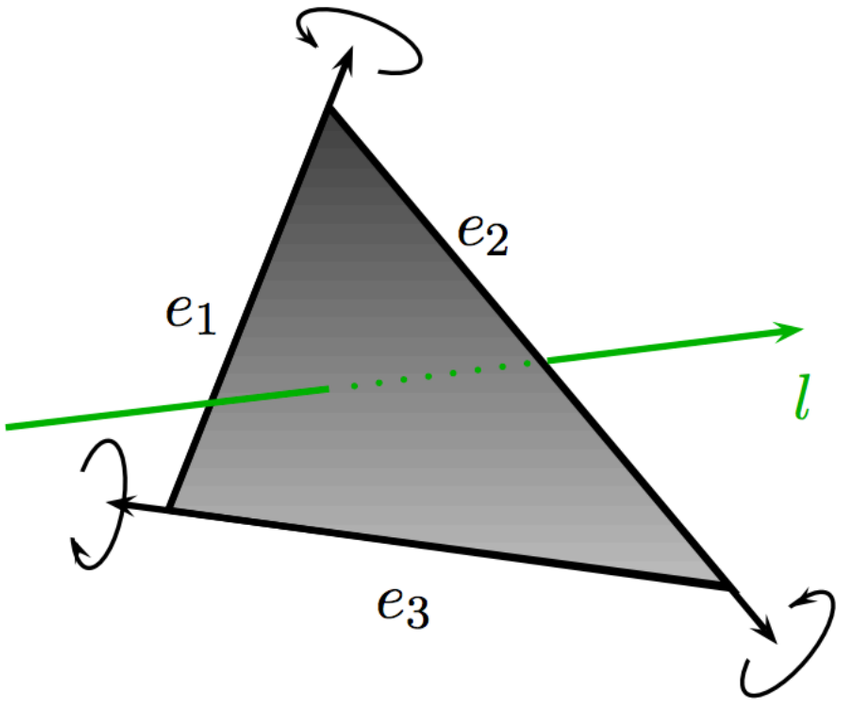
- Et donc on retrouve bien la dimension 4 de notre problème de visibilité

Espace de Plucker > droites intersectant un triangle

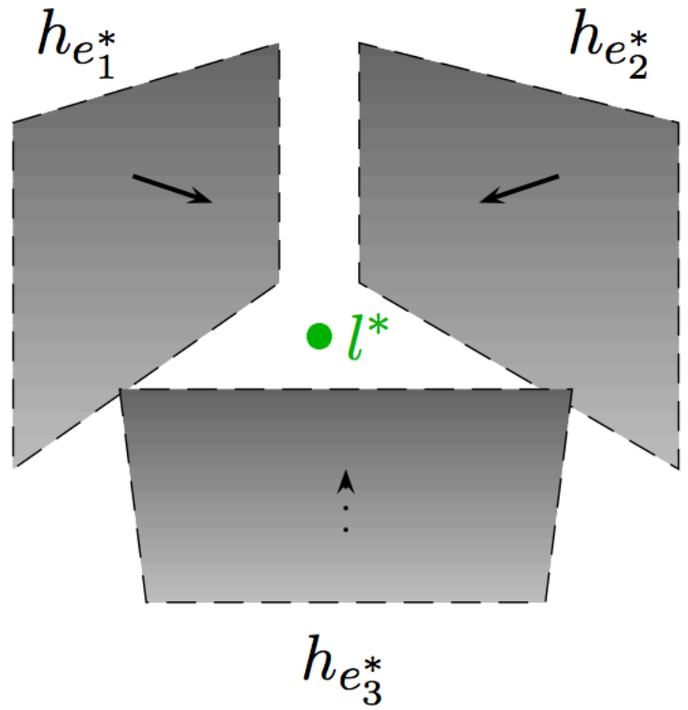


$\mathbb{R}^3$

Espace de Plucker > droites intersectant un triangle



$\mathbb{R}^3$

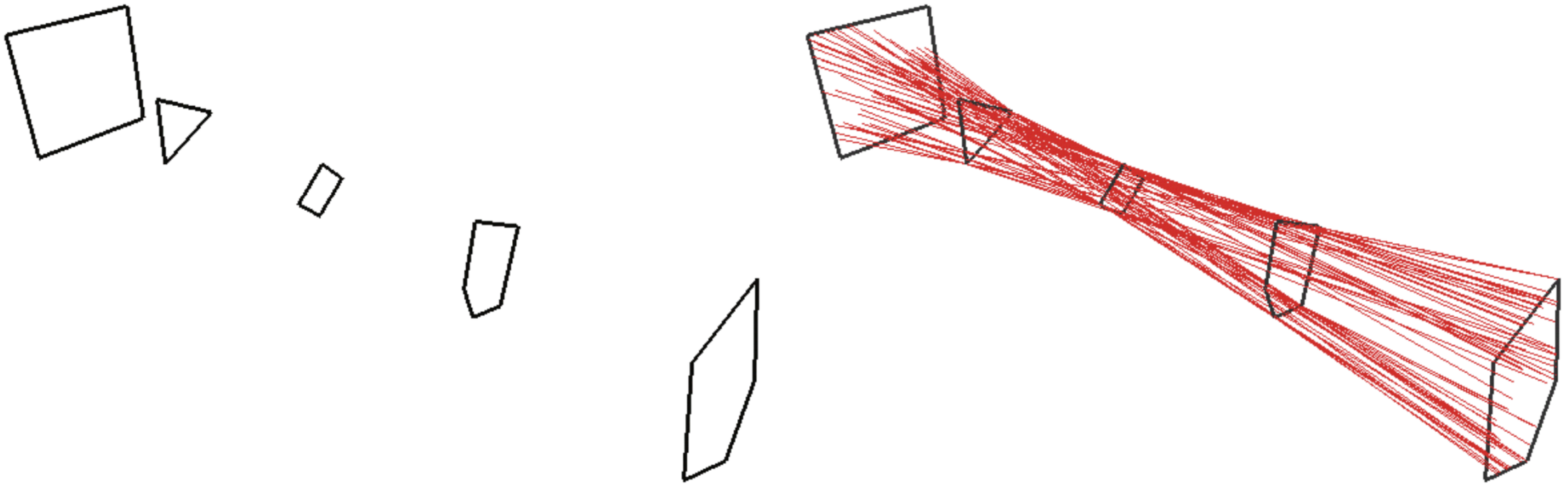


$\mathbb{P}^5$

- l a une orientation homogène avec les droites supports des arêtes du triangle
- c’est le cas pour toute droite incluse dans l’intersection des 3 demis espaces dans $\mathbb{P}^5$

Espace de Plucker > quiz

- Comment prouver qu'il existe des visibilités à travers une séquence d'ouvertures ?



Espace de Plucker > de quoi parle-t-on ?

- considérant les hp issus des arêtes de polygones
- une droite intersectant la séquence a son point de Plucker du côté + de tous les hp
- l'intersection des demis espaces hp+ contient toutes les droites intersectant la séquence

$$l^* \in \bigcap_{j=1}^{n} h^+_{e_i^*}$$

- l'intersection de demis espaces, c'est un polyèdre convexe, un polytope si fermé
- il existe des droites intersectant la sequence de polygones si :

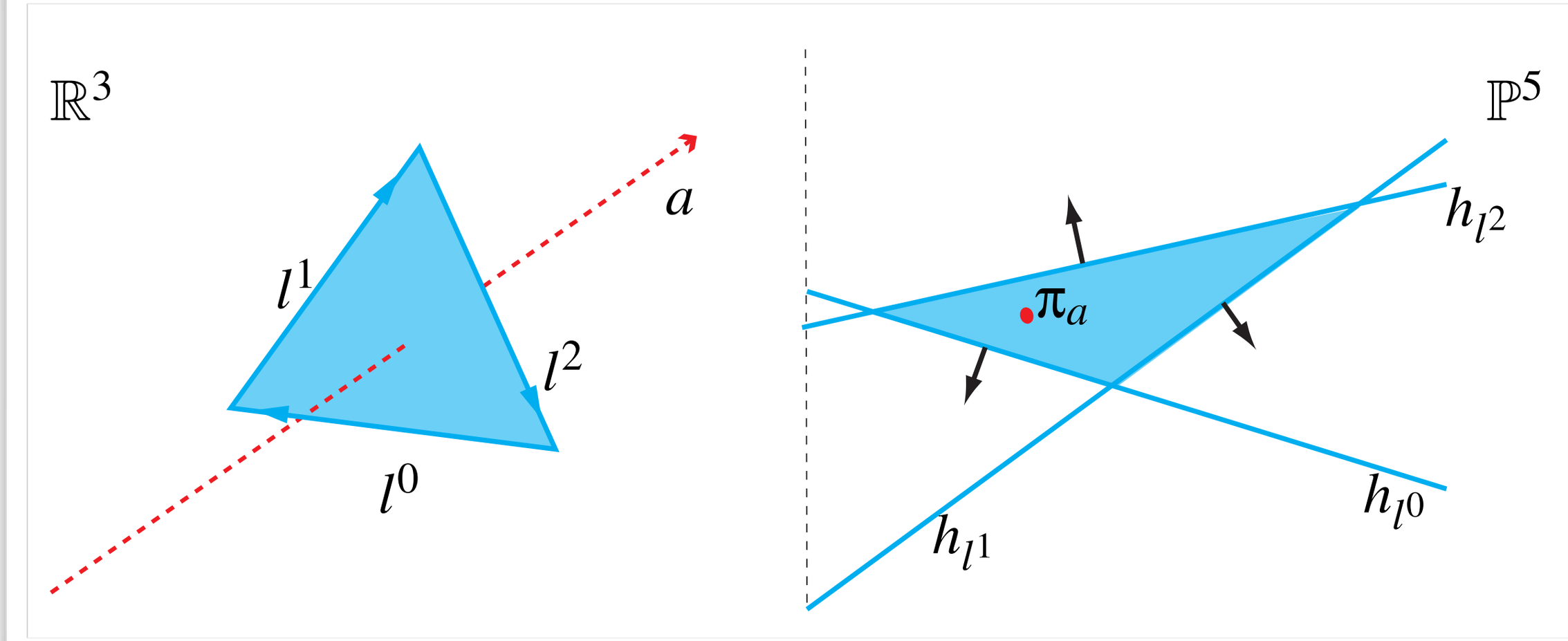
$$\bigcap_{j=1}^{n} h^+_{e_i^*} \neq \emptyset$$

Polytope ? > Alicia Boole Stott

Alicia Boole Stott (June 8, 1860 – December 17, 1940) was an Irish-English mathematician. Despite never holding an academic position, she made a number of valuable contributions to the field, receiving an honorary doctorate from University of Groningen. She is best known for coining the term "polytope" for a convex solid in four (or more) dimensions, and having an impressive grasp of four-dimensional geometry from a very early age. Alicia Boole was born in Cork, Ireland, the third daughter of mathematician and logician George Boole and Mary Everest Boole

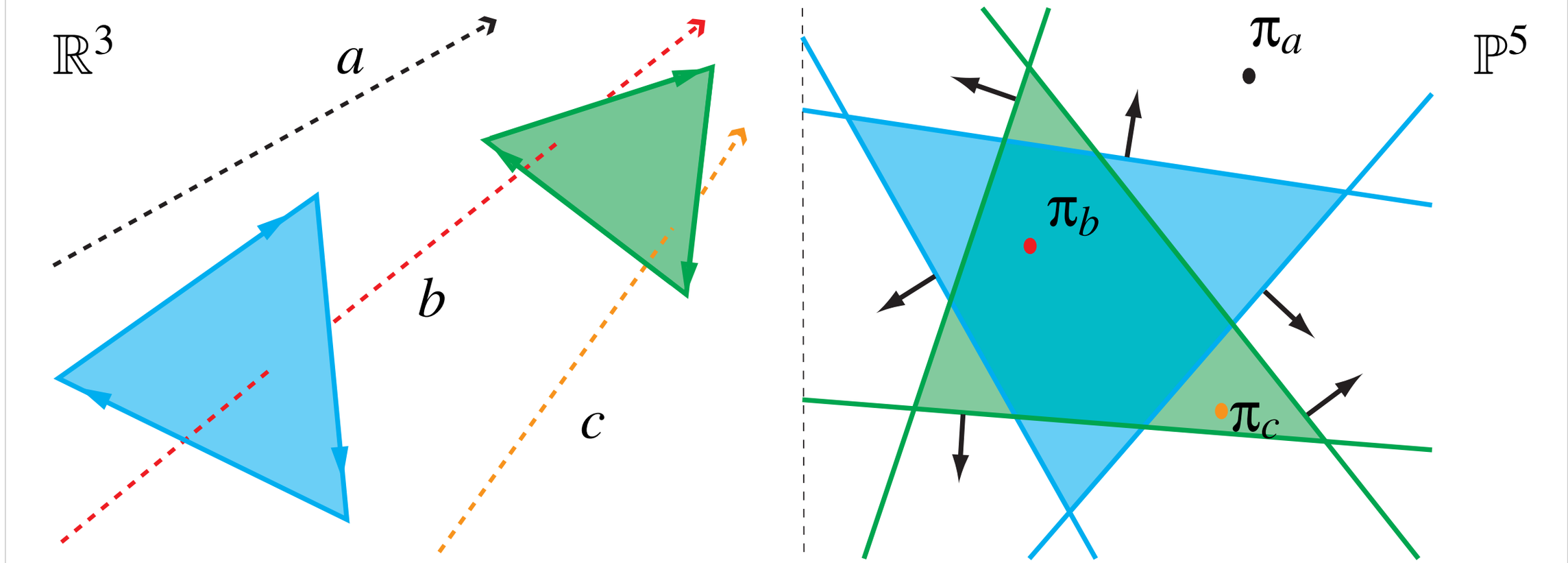
WIKIPEDIA
The Free Encyclopedia

Espace de Plucker > généralisation



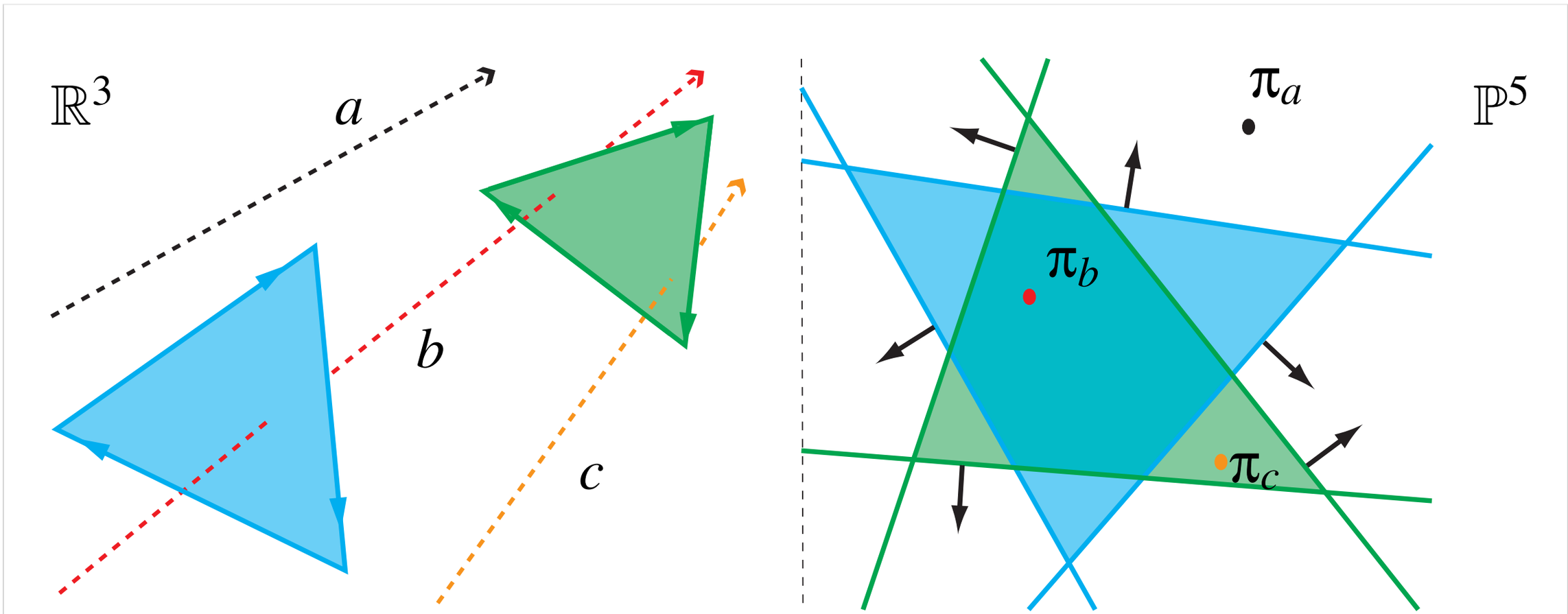
- repartons avec un seul triangle et une représentation simplifiée

Espace de Plucker > généralisation



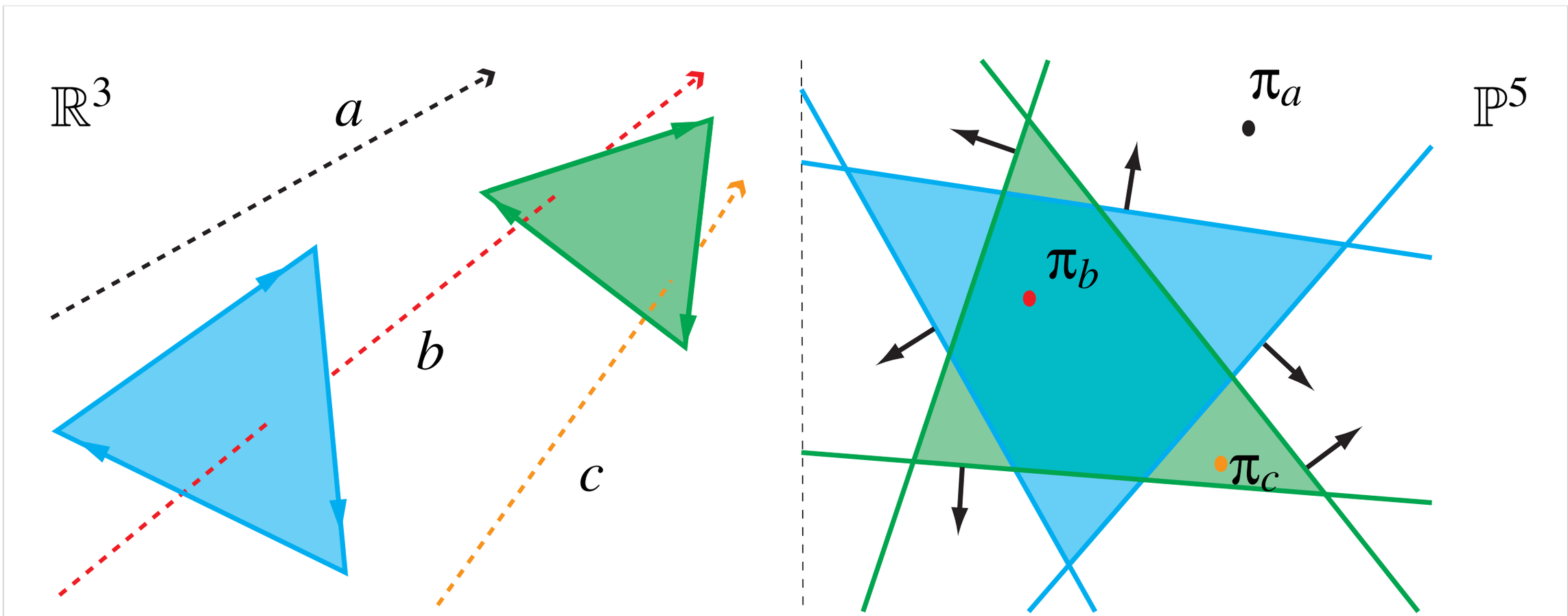
- ajoutons un autre triangle et 3 droites dans diverses configurations

Espace de Plucker > généralisation



- la représentation duale des droites supports des arêtes des triangles forme dans l'espace de Plucker un arrangement d'hyperplans

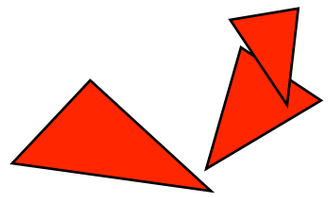
Espace de Plucker > généralisation



- cet arrangement induit une partition de l’espace de Plucker
- chaque cellule de cette partition contient des droites qui possèdent les mêmes orientations vis à vis des droites supports des triangles

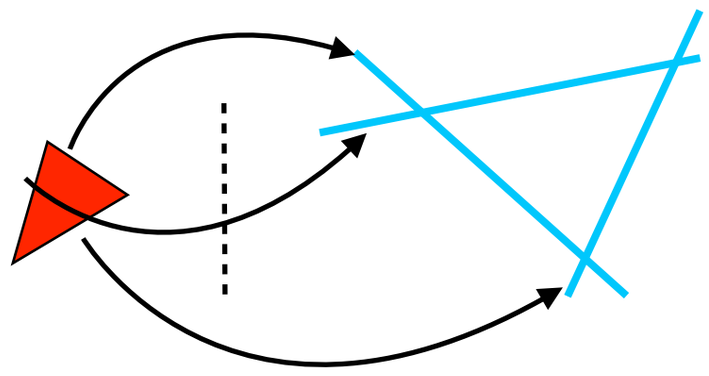
Espace de Plucker > généralisation, résumé

- étant donné des triangles



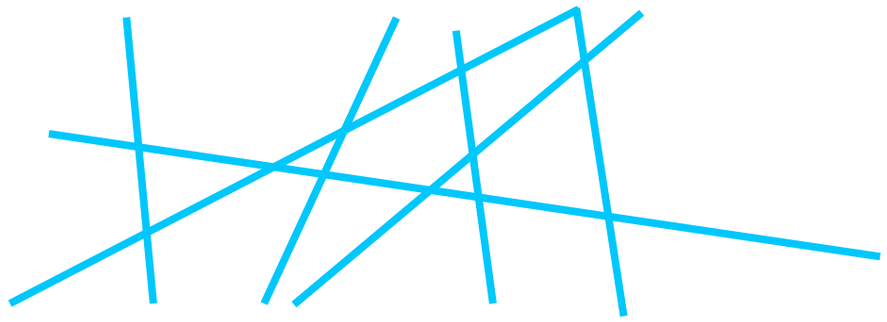
- plonger toutes les droites supports

  des arêtes sous forme d'hyperplans



- cela définit un arrangement d'hyperplans

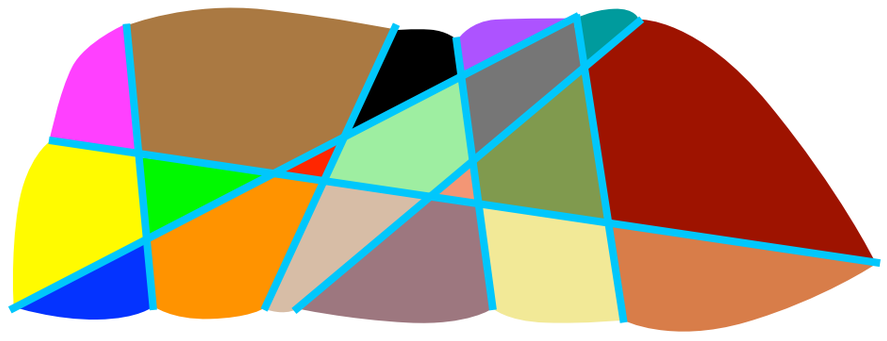
  dans l'espace de Plucker



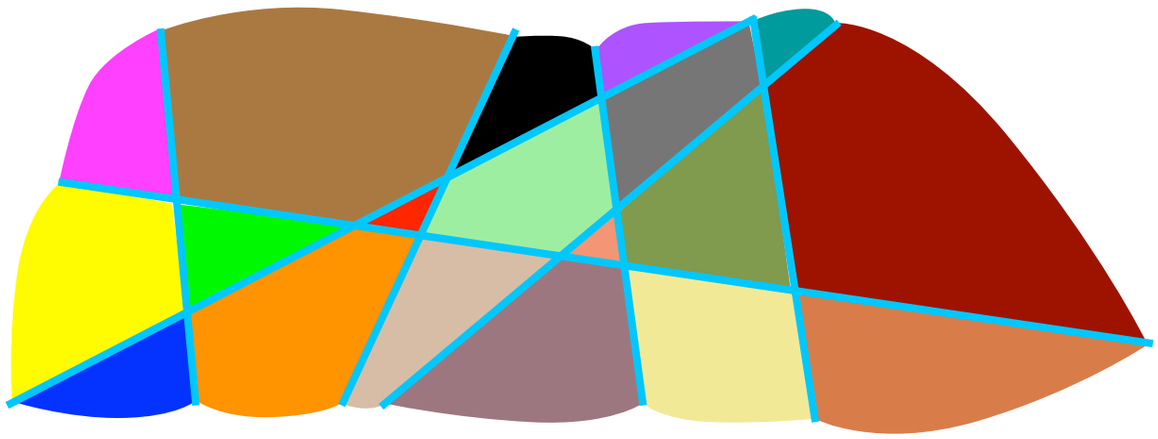
- chaque cellule représente une classe

  d'équivalence, regroupant les droites qui

  intersectent le même sous ensemble de triangles

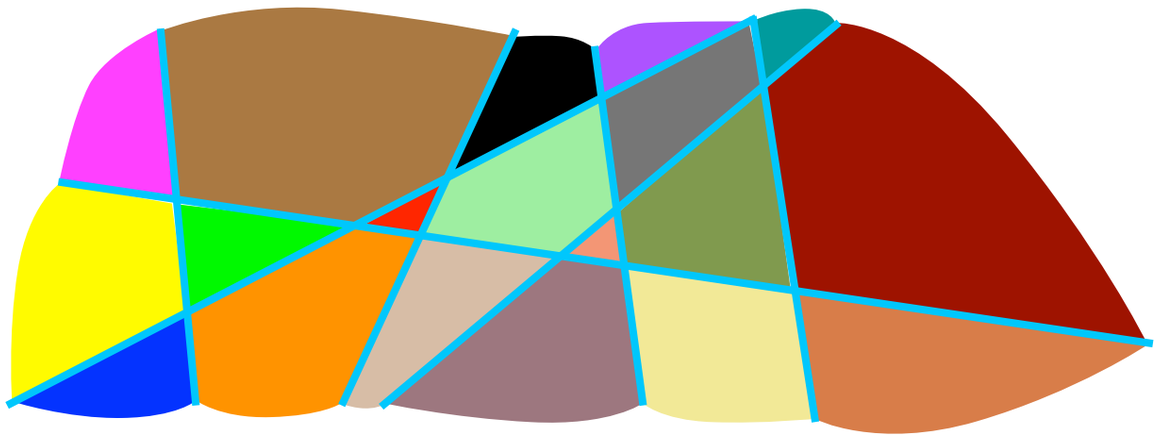


Espace de Plucker > fin de l’histoire (?)



- on dispose donc d’une description générale et analytique de la visibilité !
- ceci étant, quelle est la complexité d’un arrangement de n hyperplans dans un espace de dimension d ?
  - autrement dit, combien de cellules peuvent former n hyperplans ?

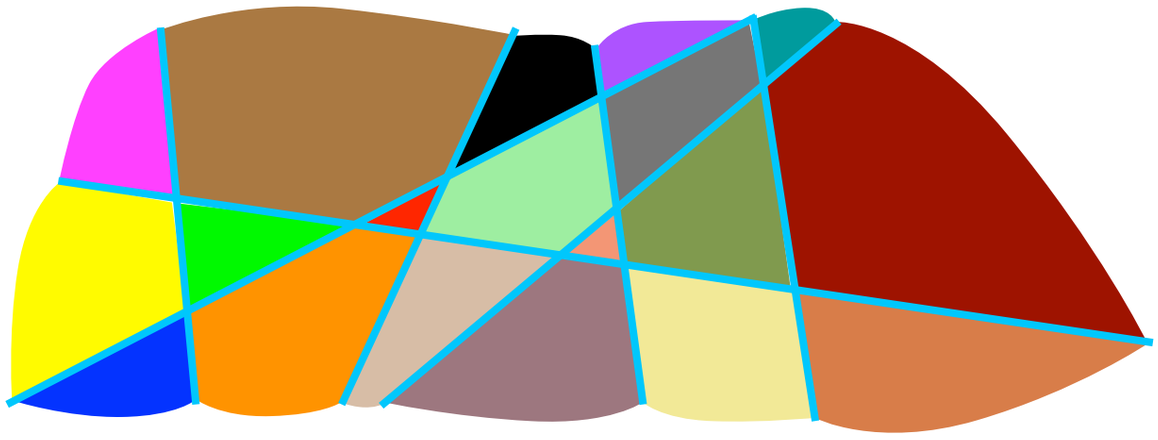
Espace de Plucker > fin de l'histoire (?)



- on dispose donc d'une description générale et analytique de la visibilité !
- ceci étant, quelle est la complexité d'un arrangement de n hyperplans dans un espace de dimension d ?
  - autrement dit, combien de cellules peuvent former n hyperplans ?

$O(n^d)$ $O(n^5)$ $O(n^4 \log(n))$

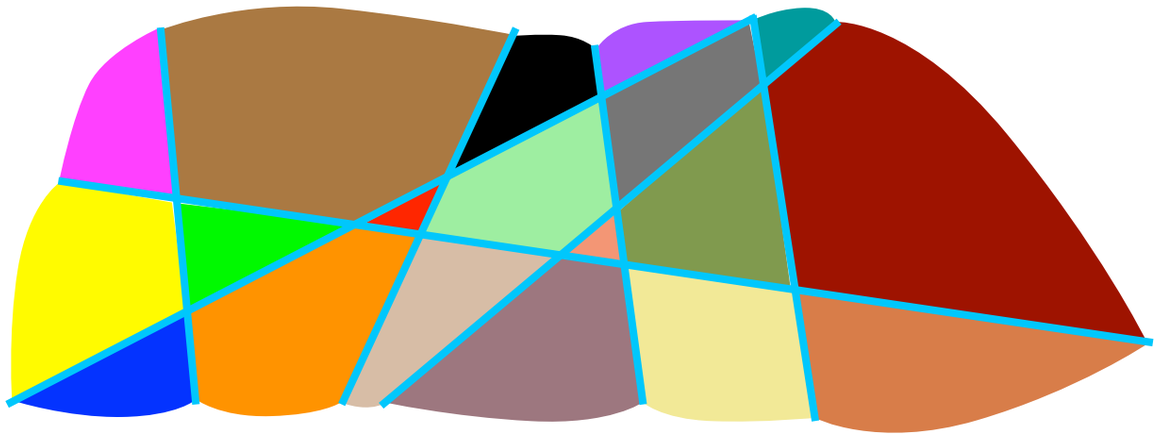
Espace de Plucker > fin de l'histoire (?)



- $O(n^4 \log(n))$ ça pique un peu…
  - soit une scène de 100 triangles, donc 300 hyperplans : plus de 37 milliards…

- algorithmiquement parlant, c'est une complexité calculatoire et mémoire

Espace de Plucker > fin de l'histoire (?)

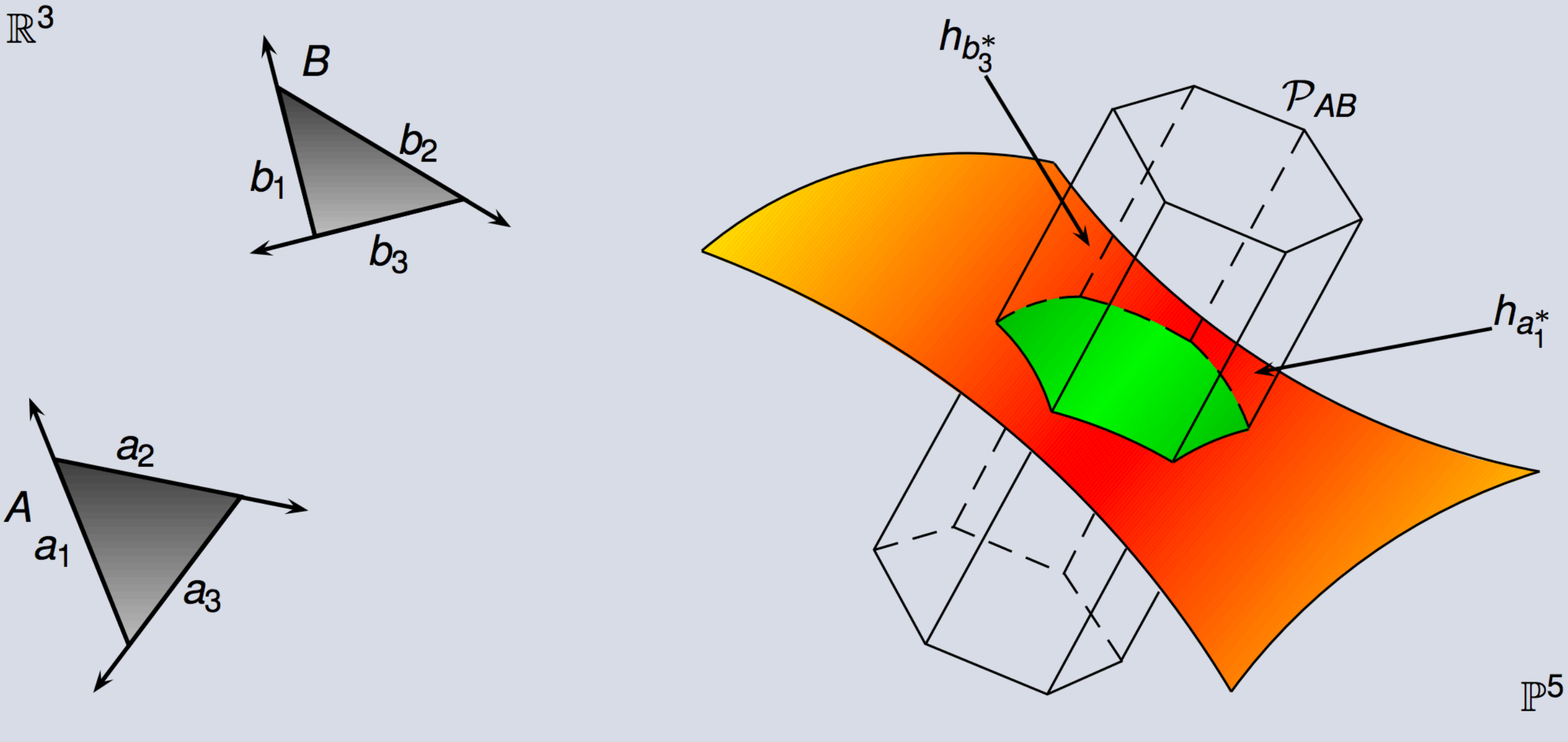


- $O(n^4 \log(n))$ ça pique un peu…
  - soit une scène de 100 triangles, donc 300 hyperplans : plus de 37 milliards…

- algorithmiquement parlant, c'est une complexité calculatoire et mémoire

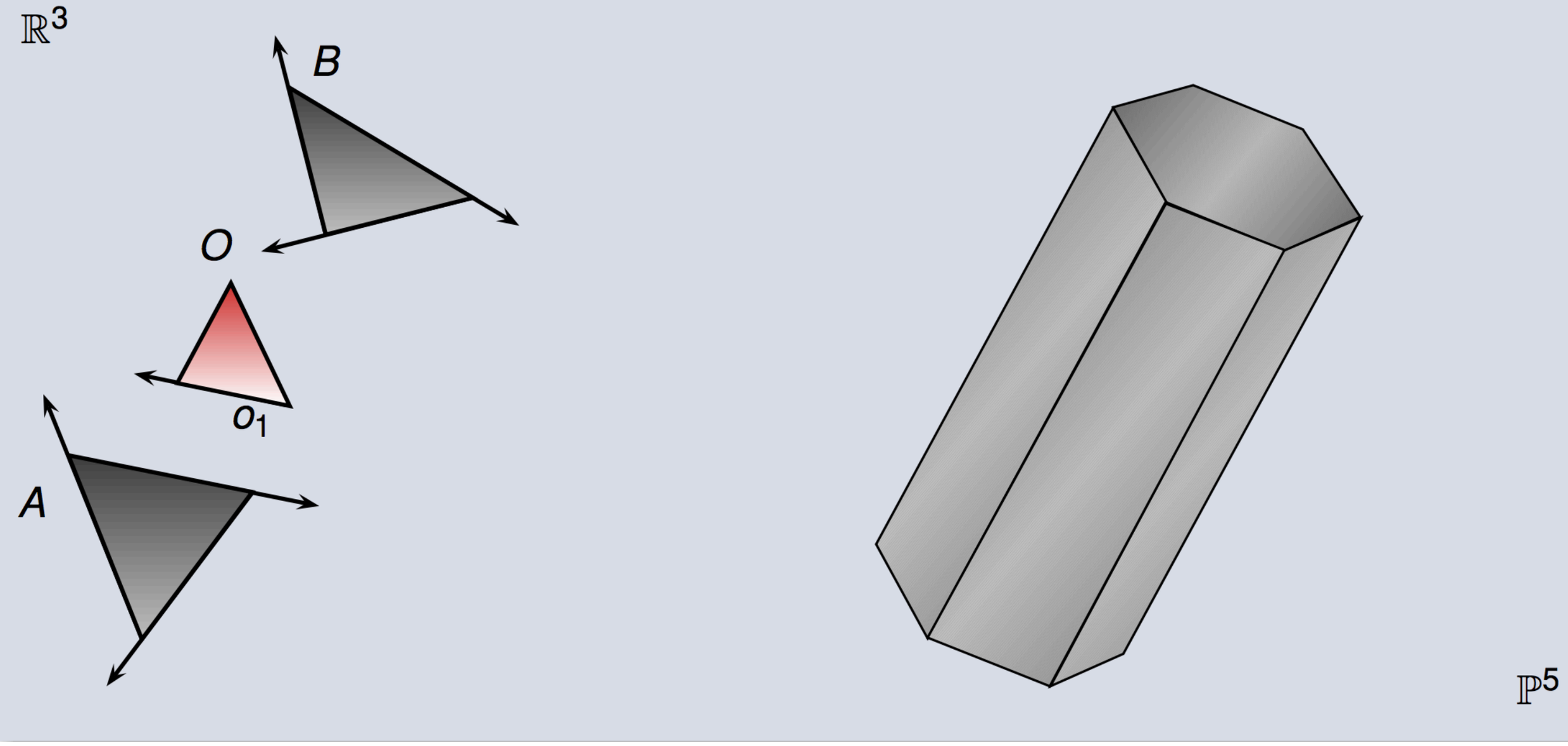
- mais, c'est une complexité « au pire »
  - positions quelconques des hp versus géométrie ordonnée des scènes 3D
  - on n'est peut être pas obligé de tout calculer à la fois

Espace de Plucker > visibilité polygone à polygone



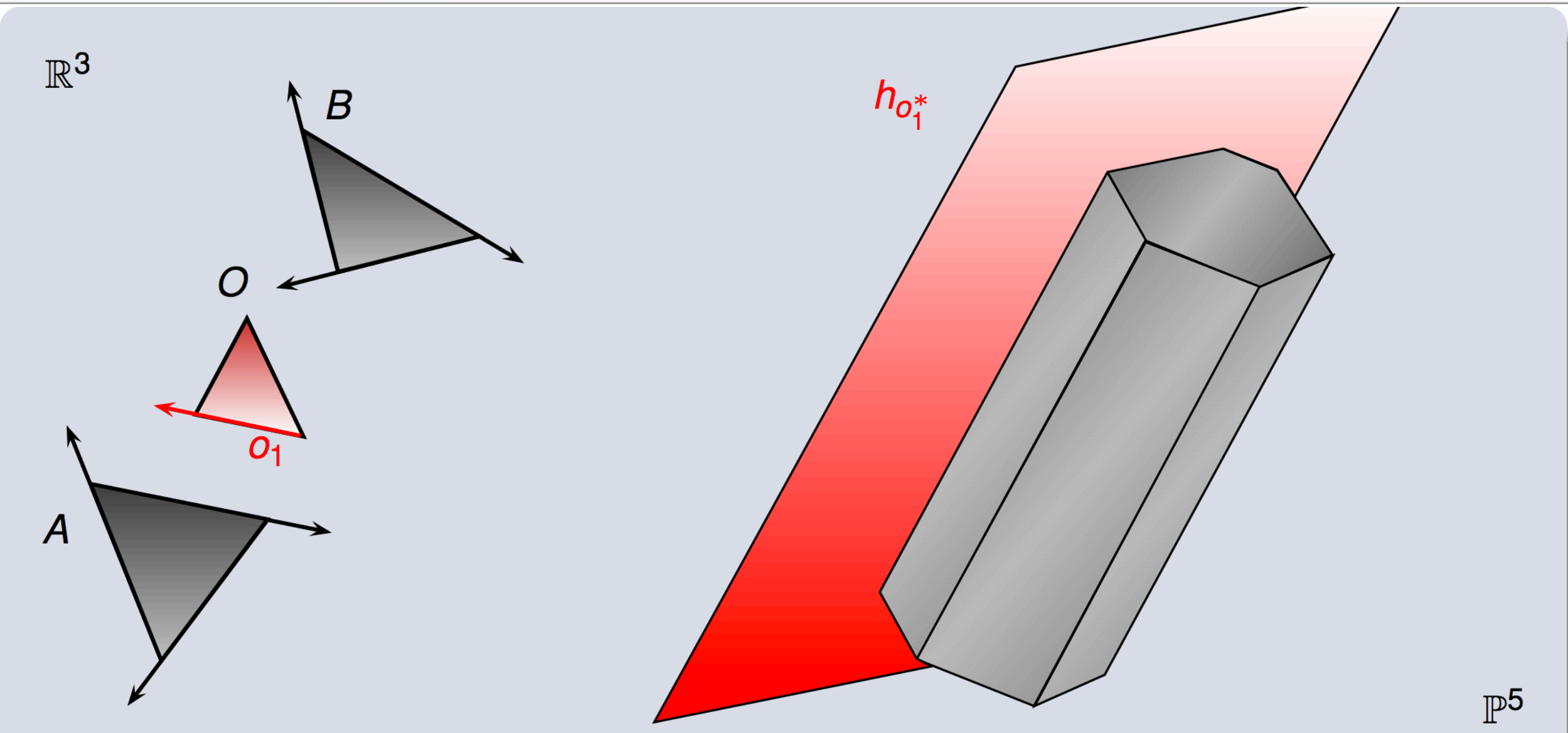
- construction d'un polytope contenant les droites intersectant A et B

Espace de Plucker > visibilité polygone à polygone



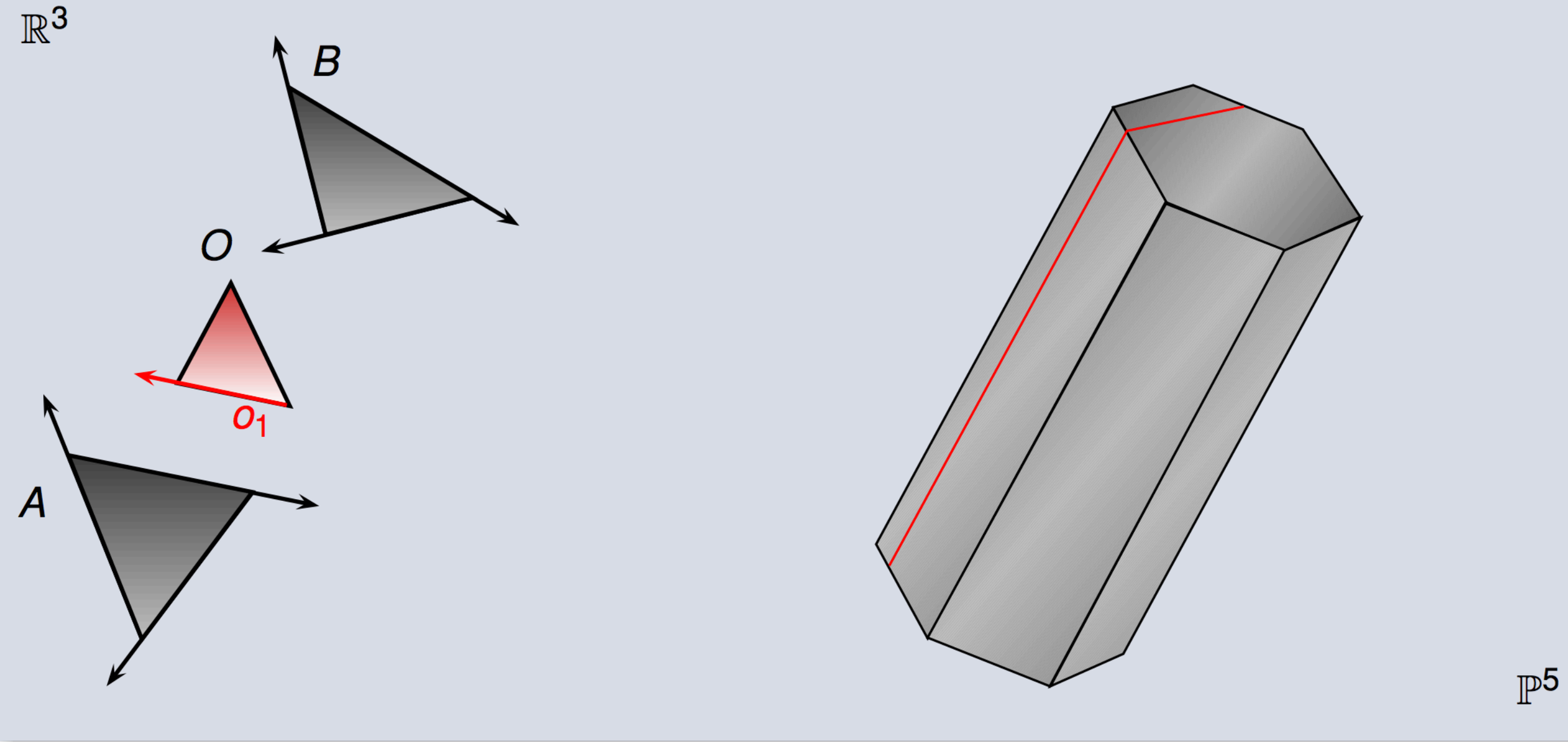
- comment prendre en compte un bloqueur O qui nuit à la visibilité de A et B ?

Espace de Plucker > visibilité polygone à polygone

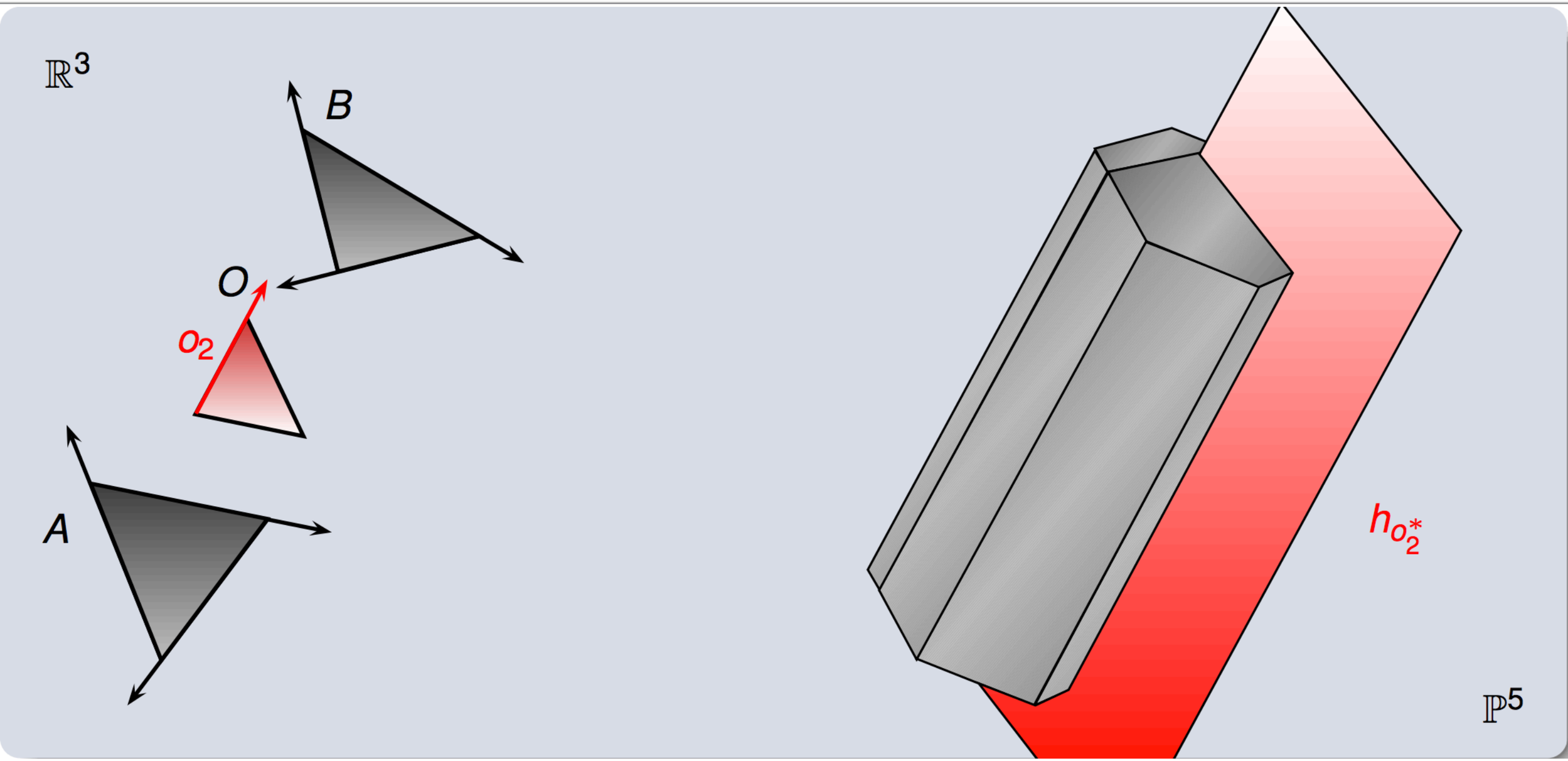


- premier hp issu de la droite support de la première arête du bloqueur
- intersection du polytope par l'hyperplan

Espace de Plucker > visibilité polygone à polygone

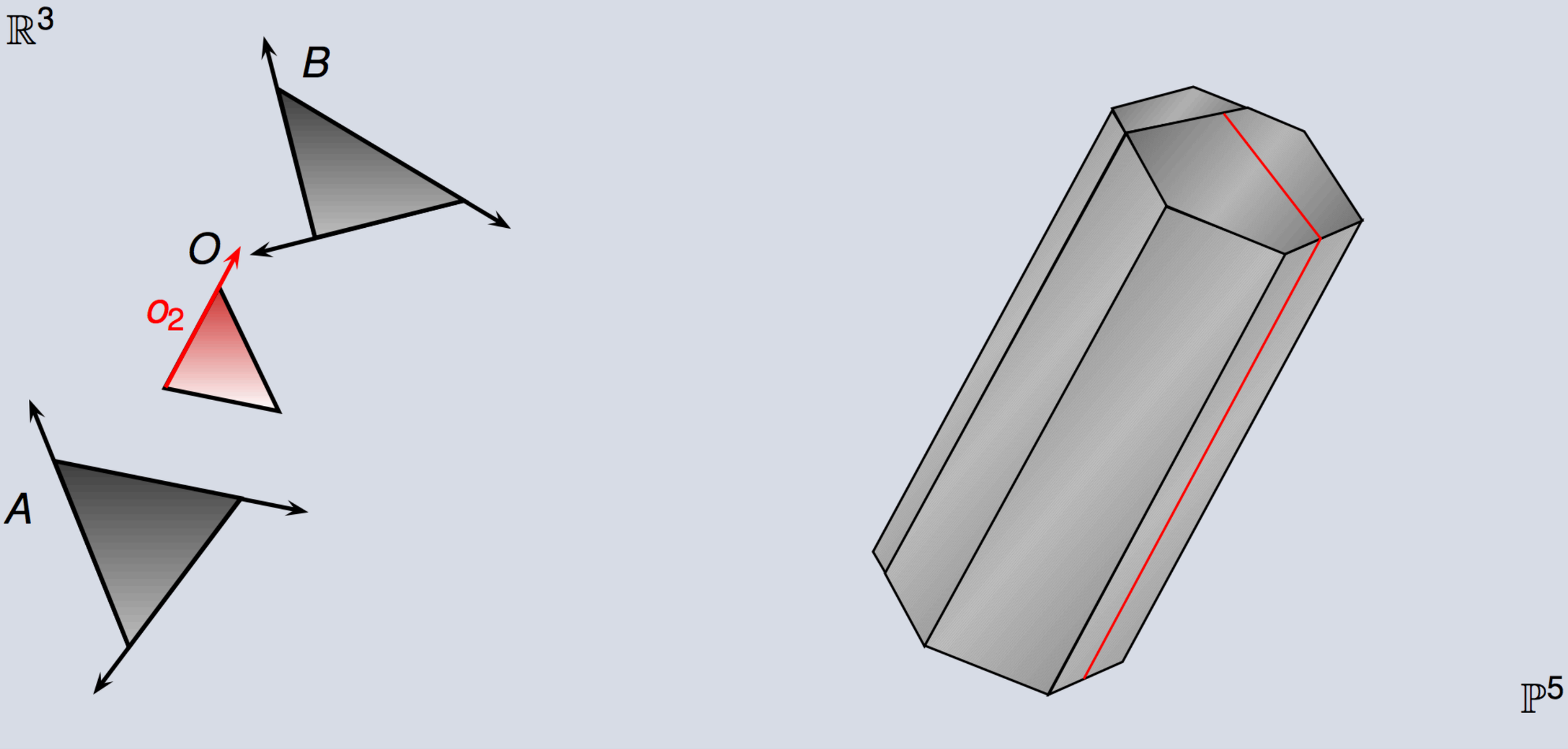


Espace de Plucker > visibilité polygone à polygone

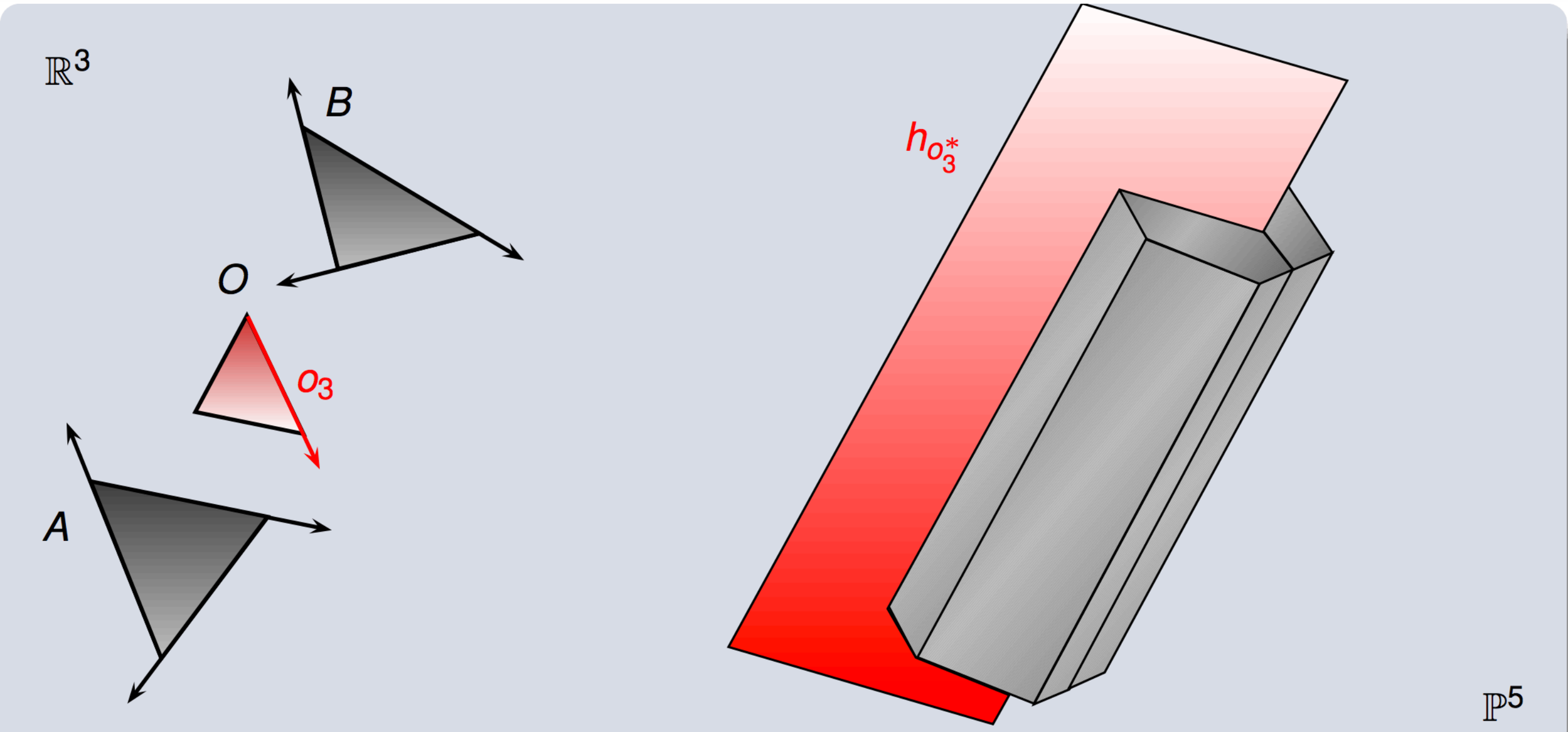


- second hp issu de la droite support de la seconde arête du bloqueur
- intersection du polytope par l'hyperplan

Espace de Plucker > visibilité polygone à polygone

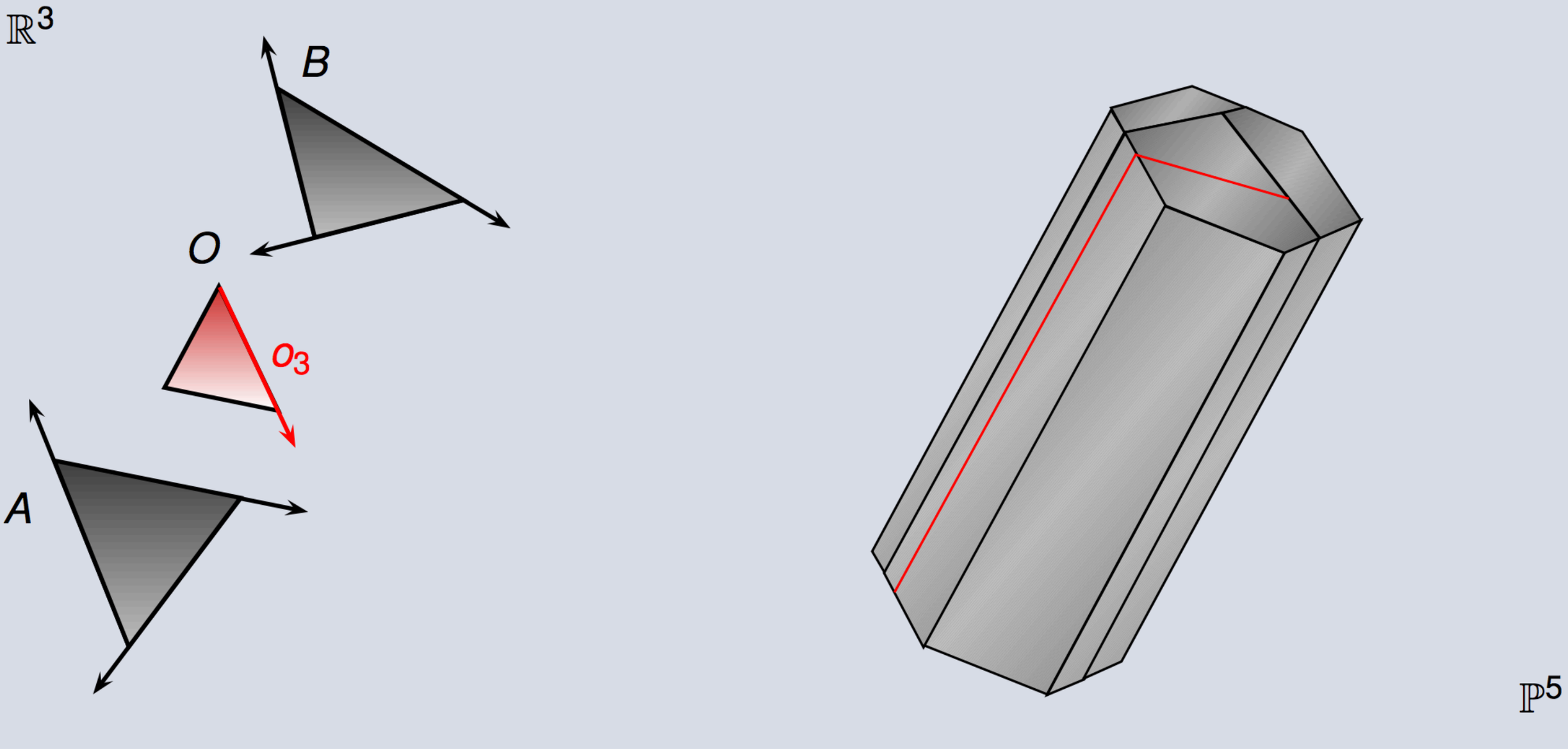


Espace de Plucker > visibilité polygone à polygone

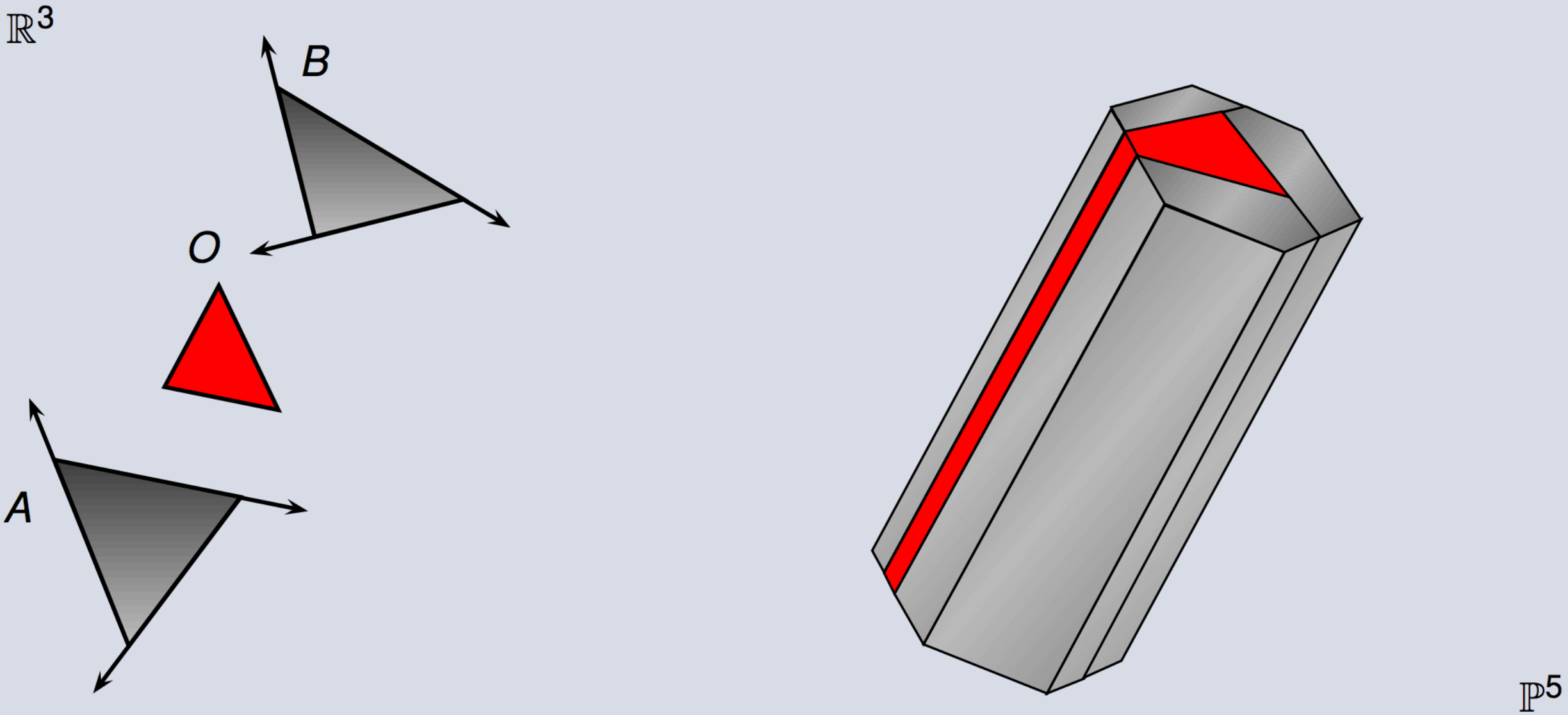


- troisième hp issu de la droite support de la troisième arête du bloqueur
- intersection du polytope par l'hyperplan

Espace de Plucker > visibilité polygone à polygone

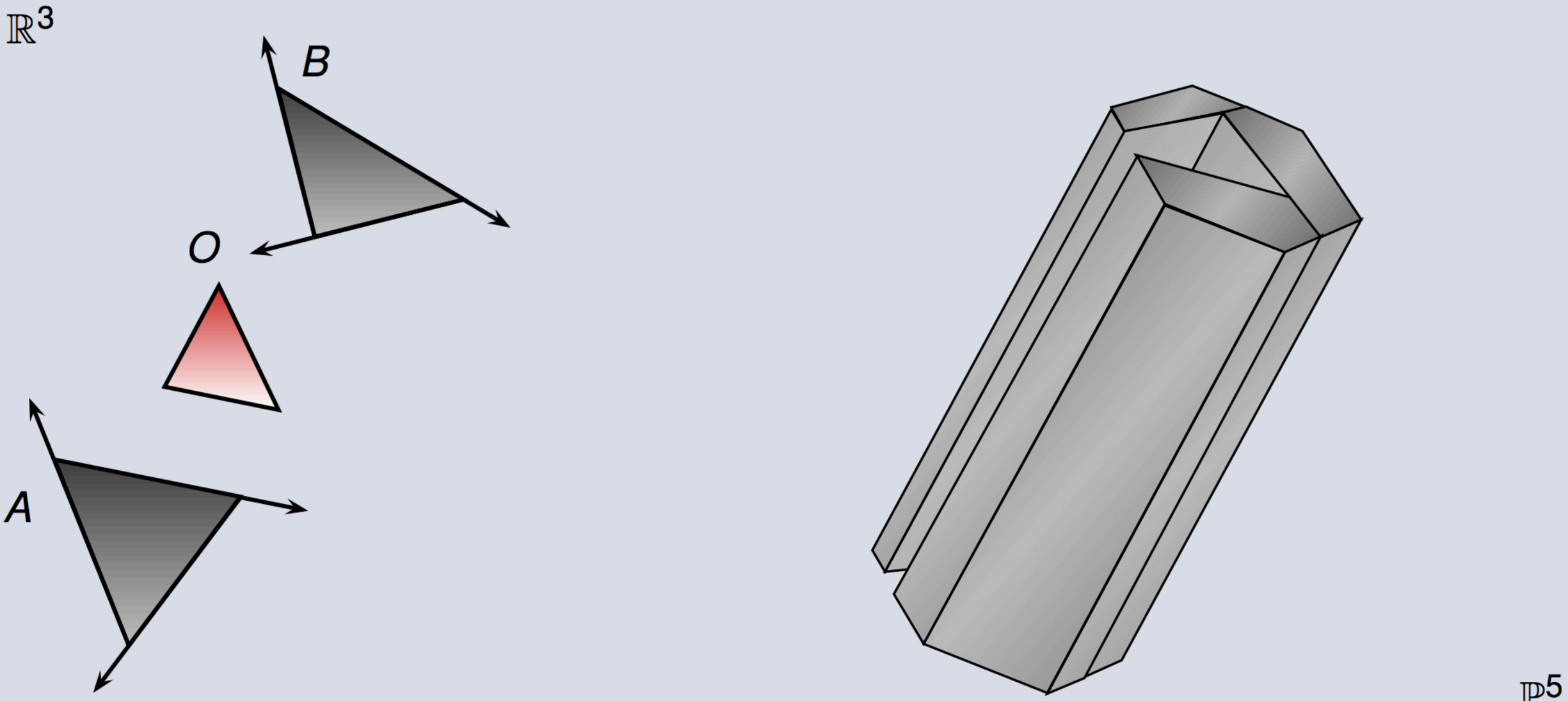


Espace de Plucker > visibilité polygone à polygone



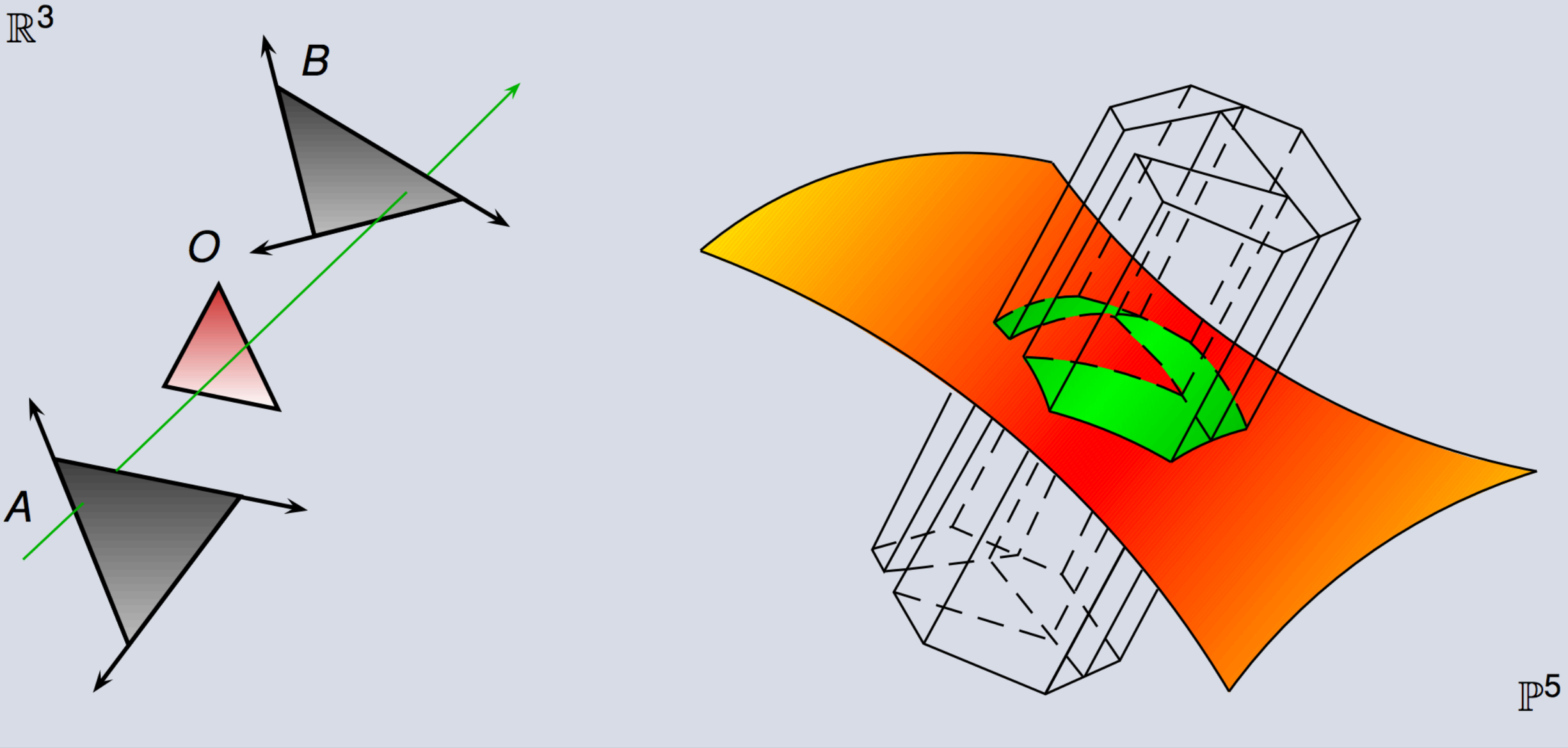
- le résultat est un complexe de polytopes
- leurs intersections avec la quadrique de Plucker forment les classes d'équivalence
- polytope « rouge » : les droites intersectant A, O, B

Espace de Plucker > visibilité polygone à polygone



- suppression des droites occultées
- on a calculé $\mathcal{P}_{AB} - \bigcap_1^3 h^+_{o_i^*}$

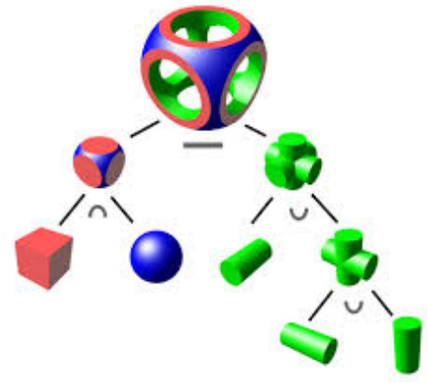
Espace de Plucker > visibilité polygone à polygone



- suppression des droites occultées
- on a calculé $\mathcal{P}_{AB} - \bigcap_1^3 h^+_{o_i^*}$

Espace de Plucker > en pratique

- ni plus ni moins que des opérations CSG
- mais en 5D…



Deux approches :

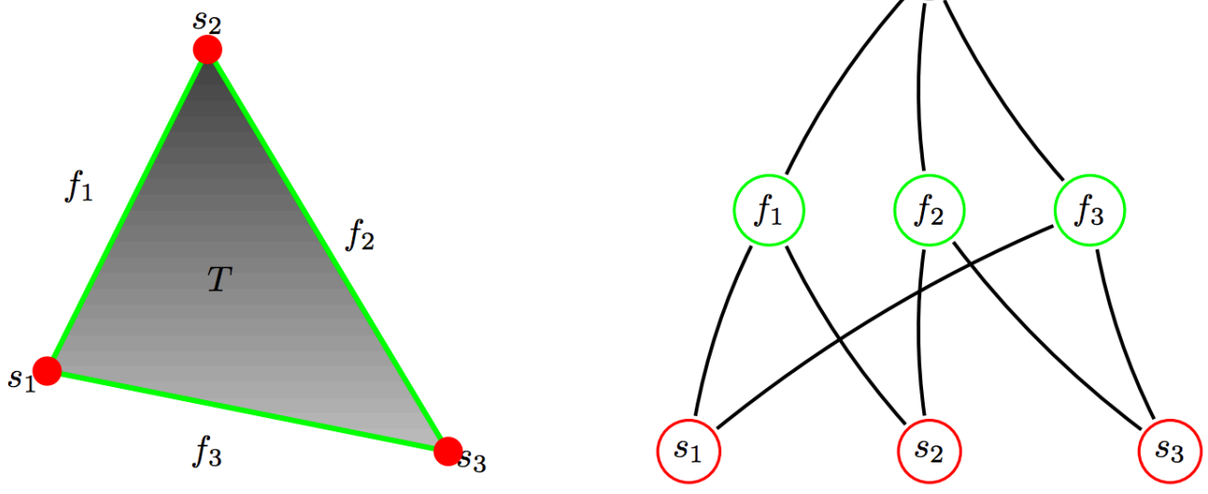
- numérique : résolution de système d’équations linéaires
- géométrique / topologique

Espace de Plucker > en pratique

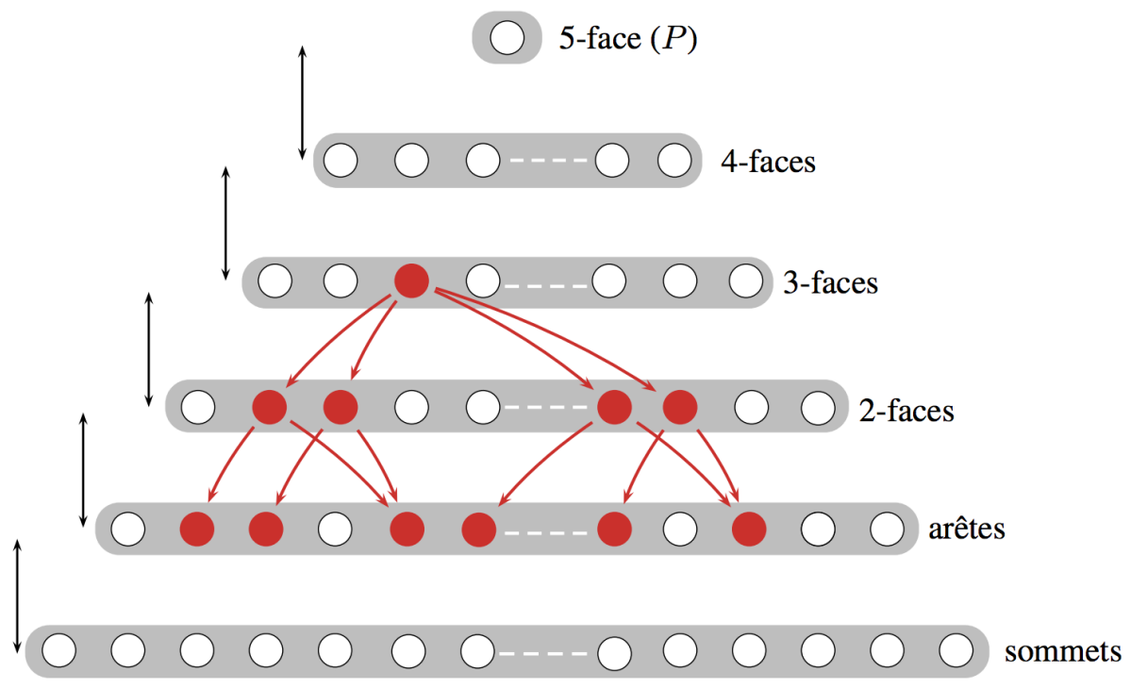
Un polytope se caractérise par

- sa H-représentation (ses hyperplans)
- sa V-représentation (ses sommets)

A partir des relations d'incidence entre la H et V-représentation, on peut calculer le graphe d'incidence d'un polytope :



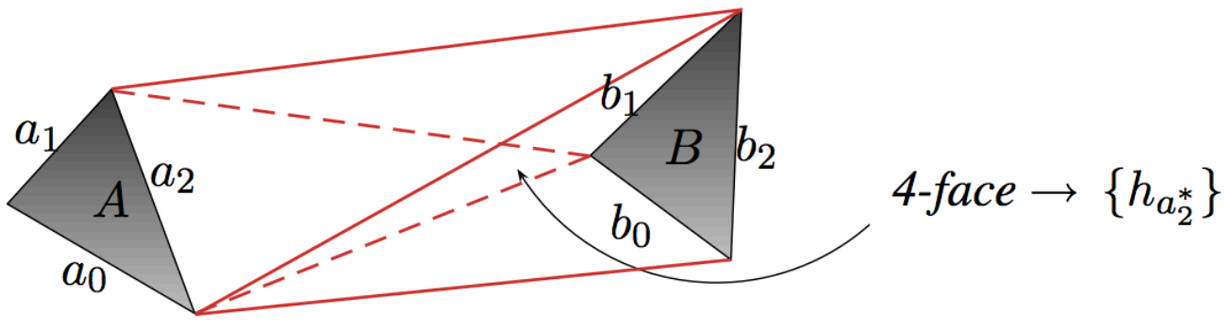
exemple en 2D

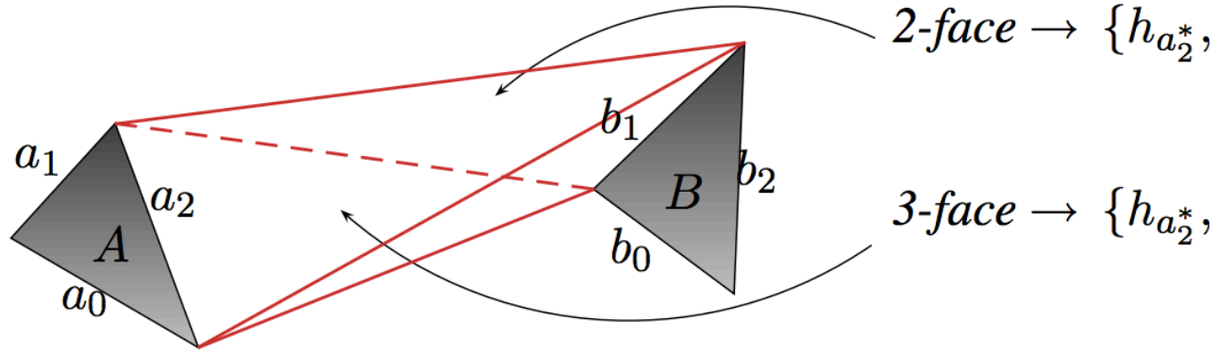


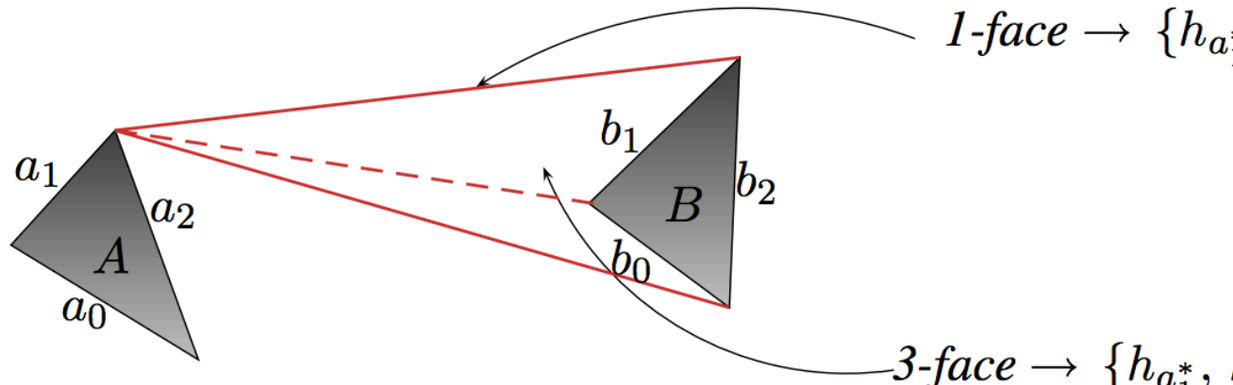
exemple en 5D

Espace de Plucker > en pratique

Faces d’un de nos polytopes et interprétation en 3D

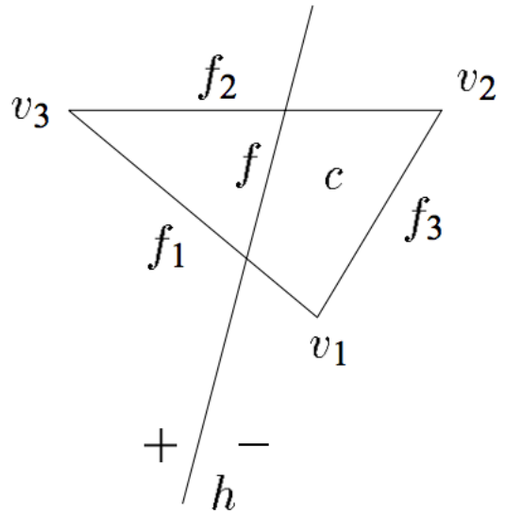


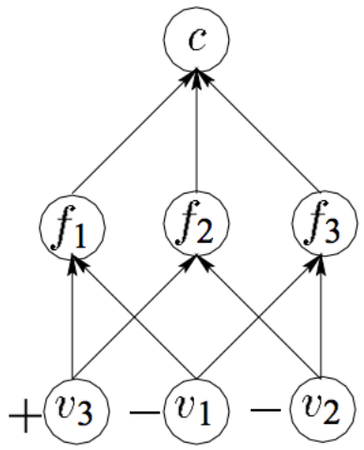


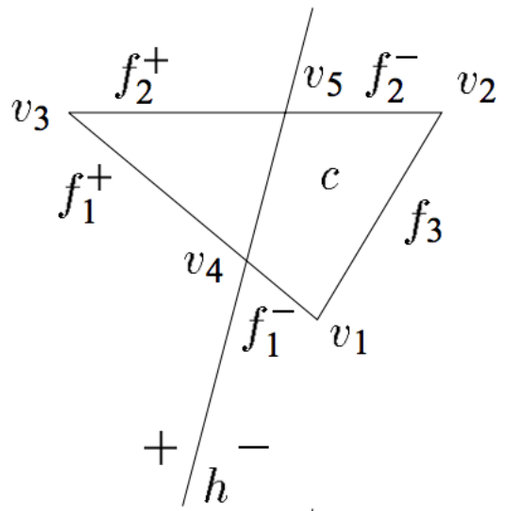


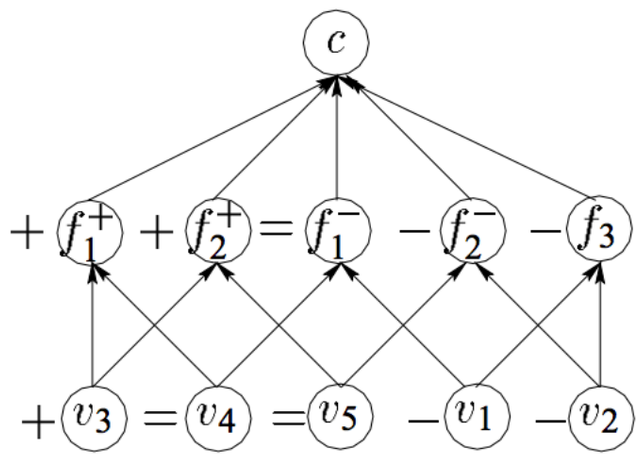
Espace de Plucker > en pratique

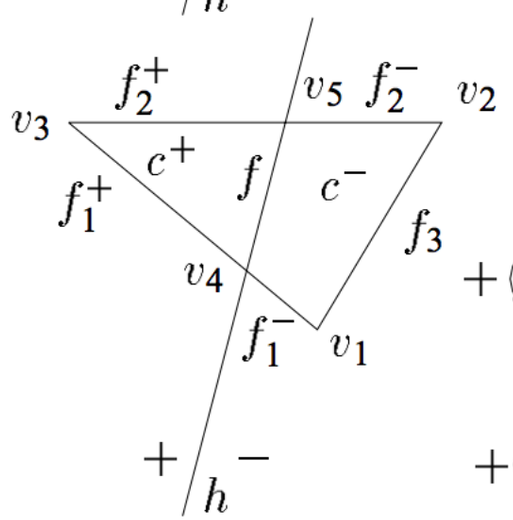
On sait ensuite calculer l'intersection de polytopes par un hyperplan en se basant uniquement sur la classification initiale de ses sommets

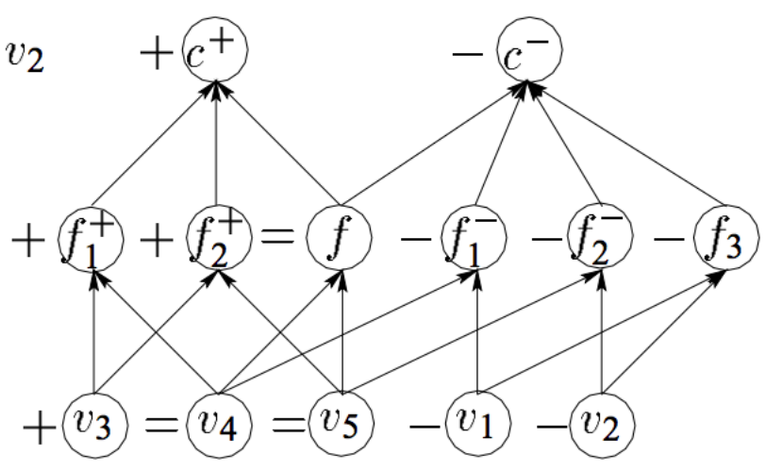








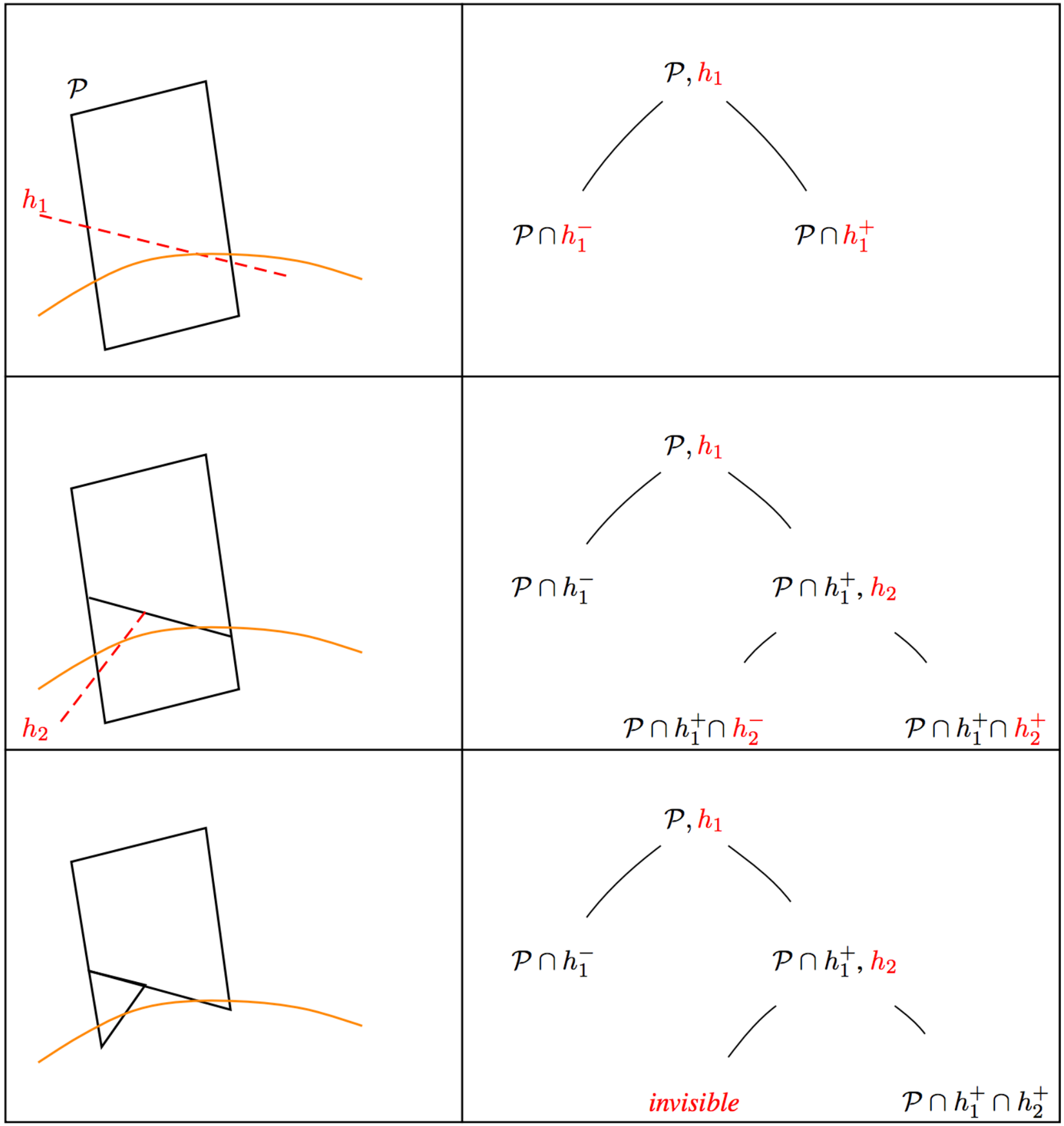




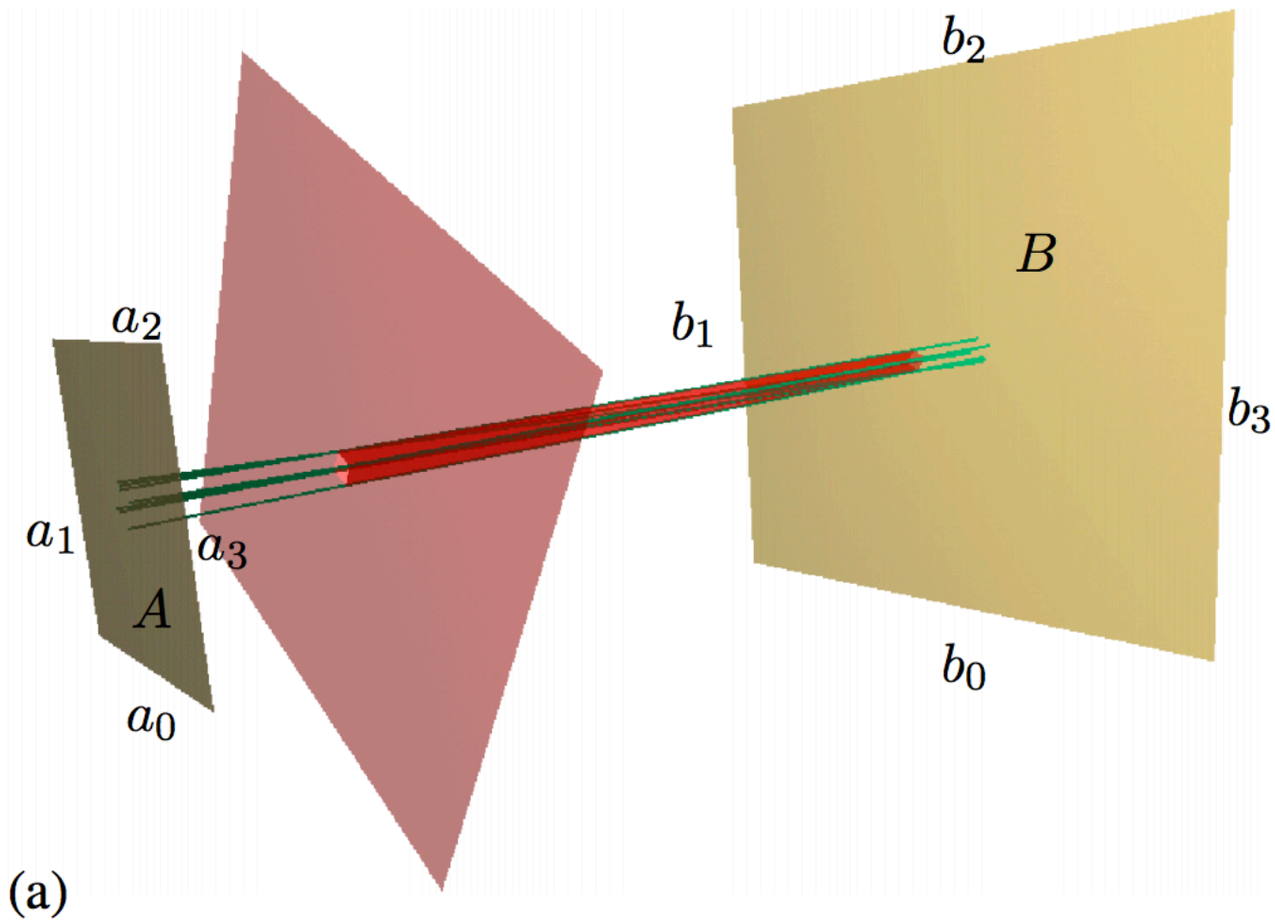
Avantages :

- plus robuste numériquement
- fonctionne en toute dimension

Espace de Plucker > en pratique



Espace de Plucker > interprétation des k-faces



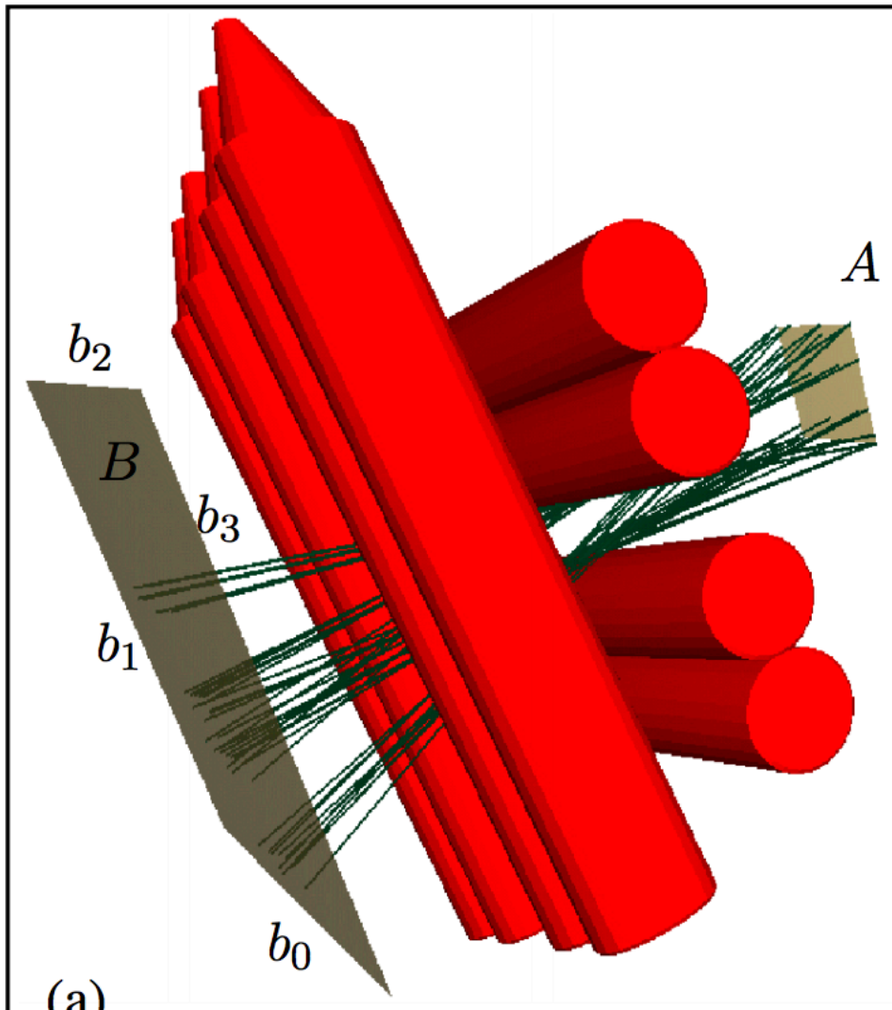
(a)

k-faces	#k-faces	$\in \mathcal{P}_{AB}$
3-faces	25	0
4-faces	8	0
5-face	1	1

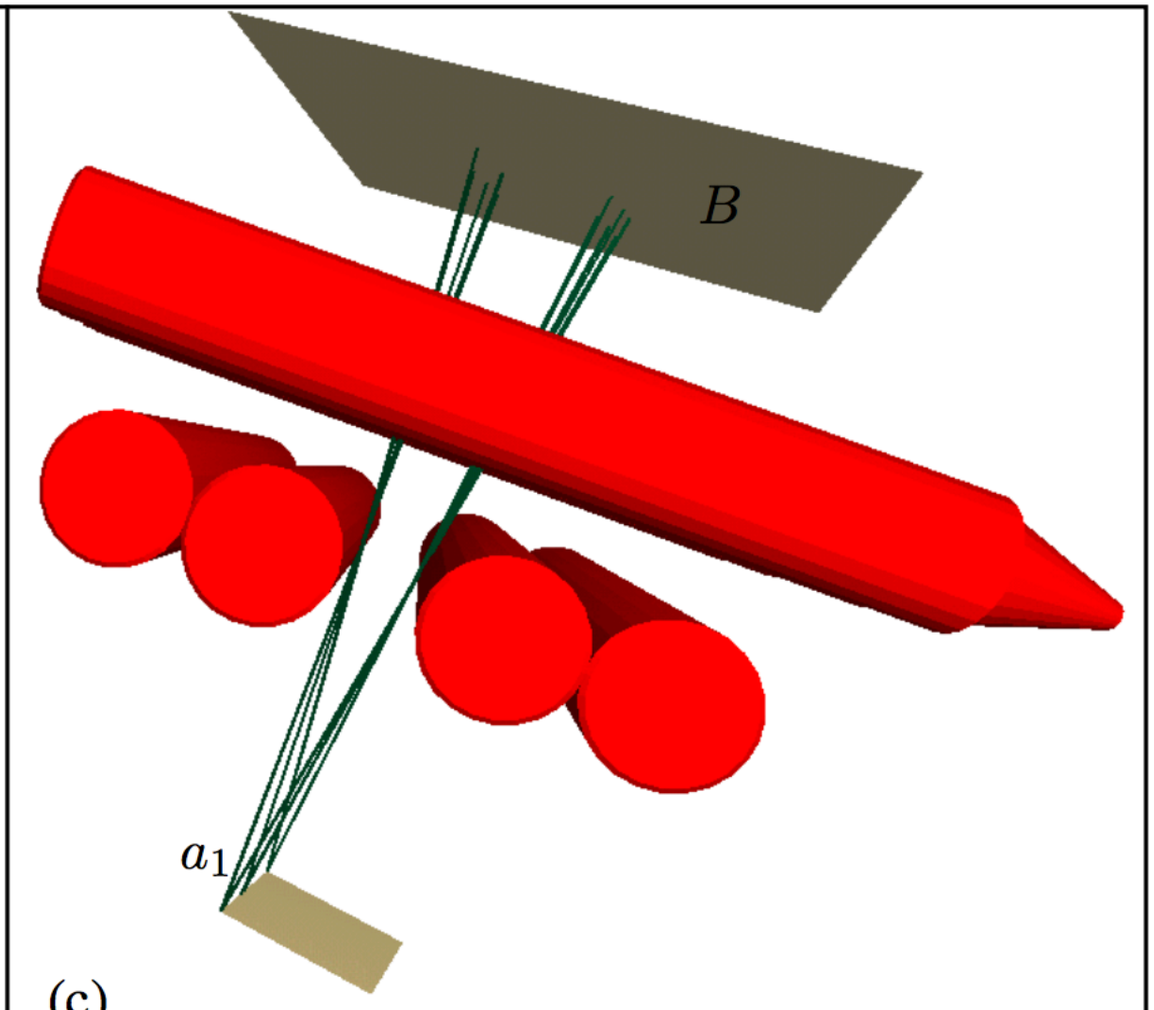
	A	a_0	a_1	a_2	a_3
B	✓	-	-	-	-
b_0	-	-	-	-	-
b_1	-	-	-	-	-
b_2	-	-	-	-	-
b_3	-	-	-	-	-

(b)

Espace de Plucker > interprétation des k-faces



(a)



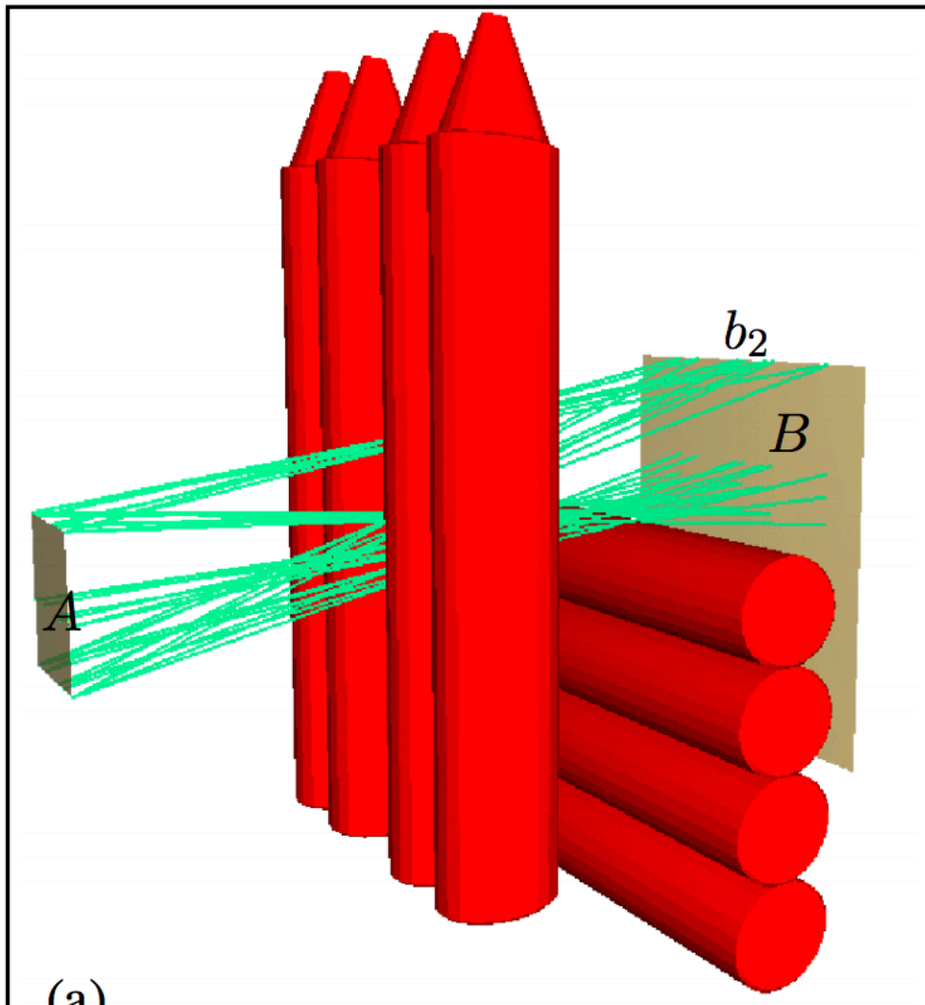
(c)

k-faces	#k-faces	$\in \mathcal{P}_{AB}$
3-faces	206	0
4-faces	27	4
5-face	1	1

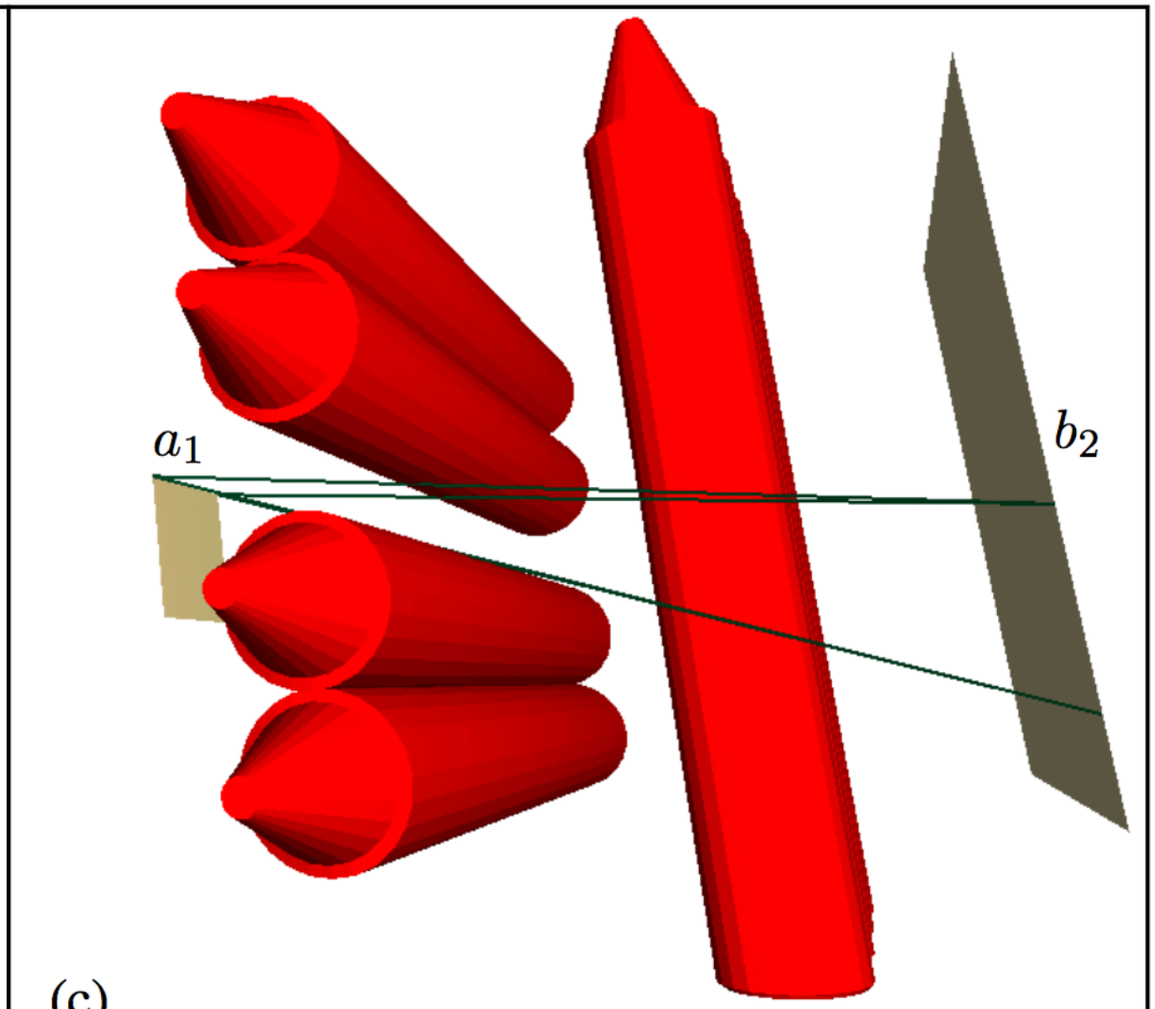
(b)

	A	a_0	a_1	a_2	a_3
B	√	√	√	√	√
b_0	-	-	-	-	-
b_1	-	-	-	-	-
b_2	-	-	-	-	-
b_3	-	-	-	-	-

Espace de Plucker > interprétation des k-faces



(a)



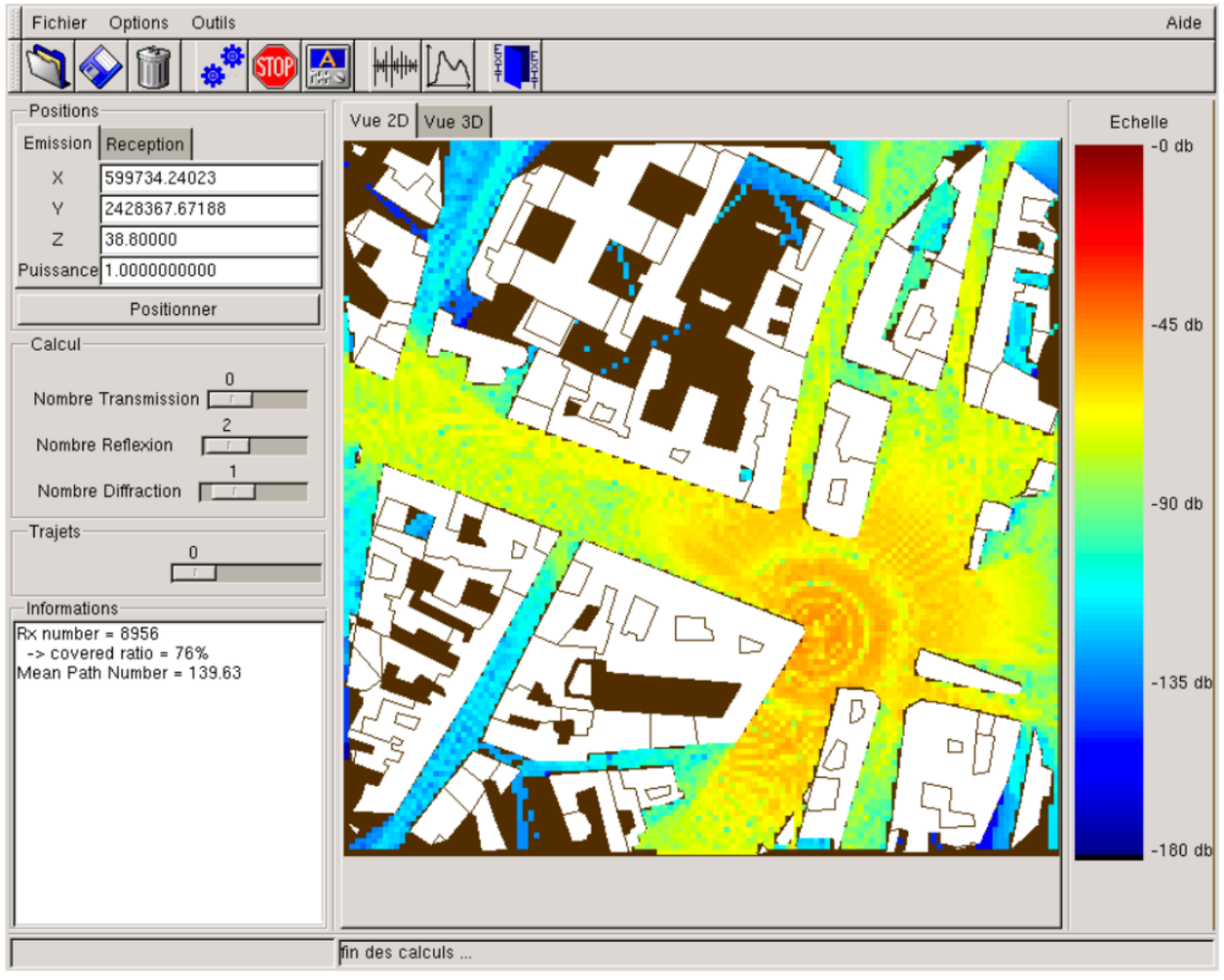
(c)

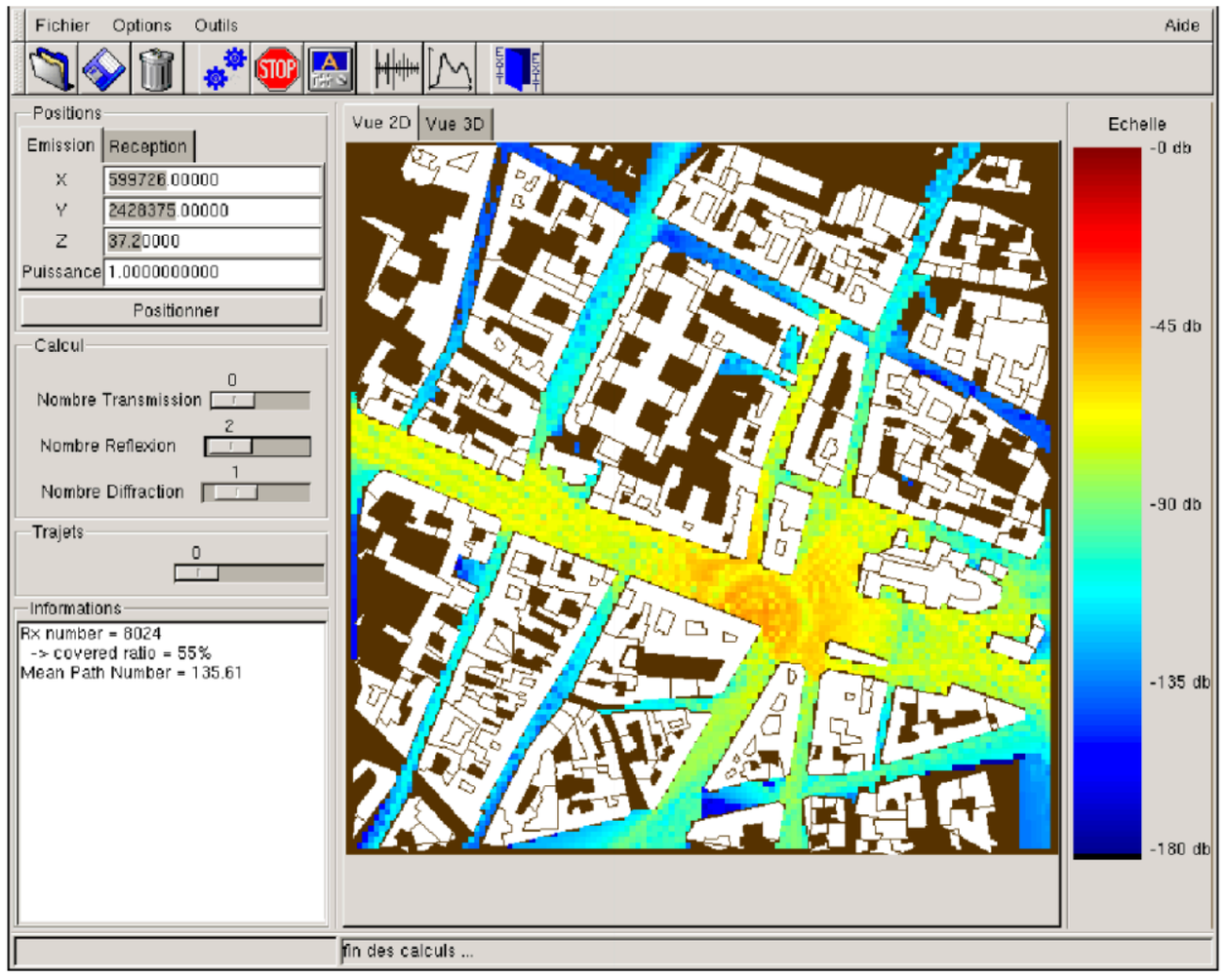
(b)

k-faces	#k-faces	$\in \mathcal{P}_{AB}$
3-faces	145	4
4-faces	22	5
5-face	1	1

	A	a_0	a_1	a_2	a_3
B	√	√	√	√	√
b_0	-	-	-	-	-
b_1	-	-	-	-	-
b_2	√	√	√	√	√
b_3	-	-	-	-	-

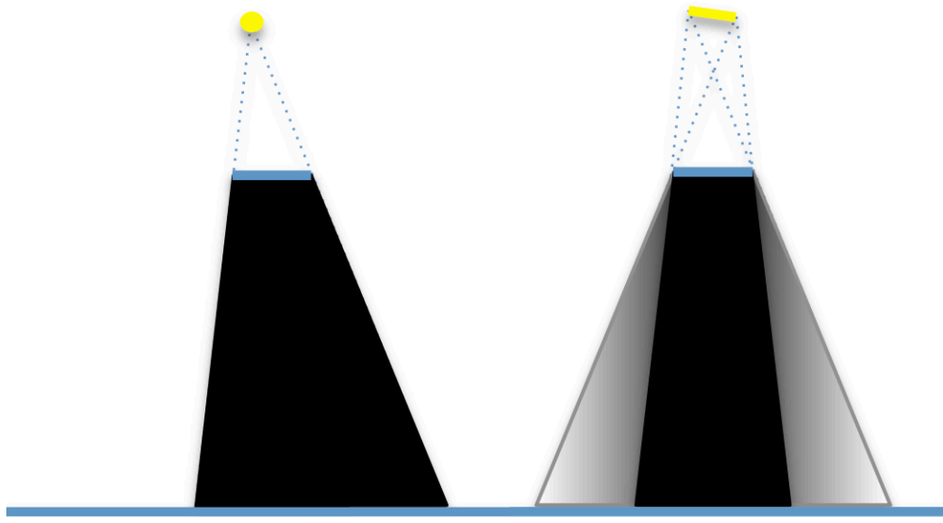
Applications > propagation électromagnétique

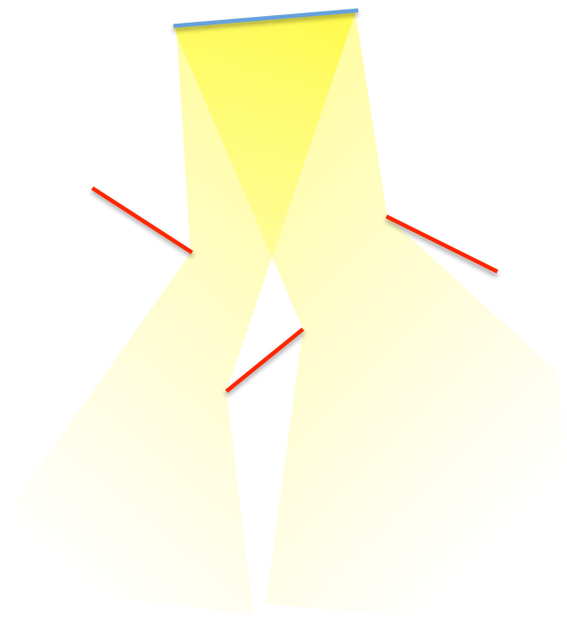


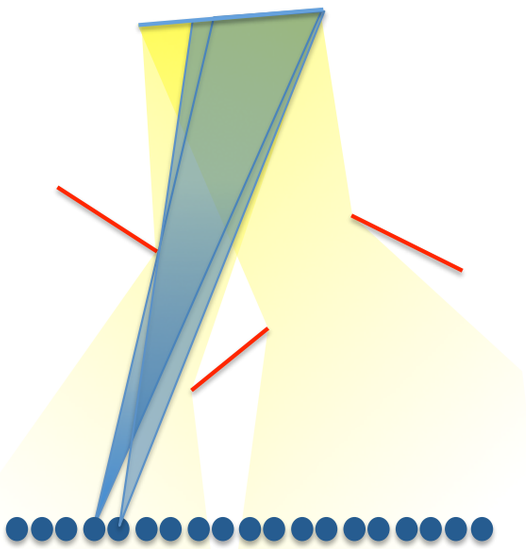


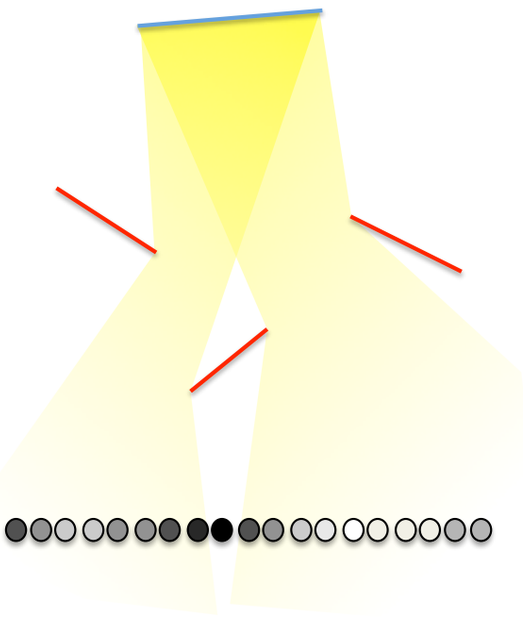
Interactions	Paris A (8983 récepteurs)		Paris B (8024 récepteurs)	
	Avec GV	Sans GV	Avec GV	Sans GV
1 R 0 D	$3s$	$10s$	$4s$	$26s$
2 R 0 D	$36s$	$2h\,55min\,24s$	$34s$	$> 1j\,8h$
2 R 1 D	$3h\,18min$	$> 3\,ans$	$7h\,16min$	-
3 R 0 D	$23min\,18s$	$> 8\,mois$	$34min\,23s$	-

Applications > ombre douces analytiques

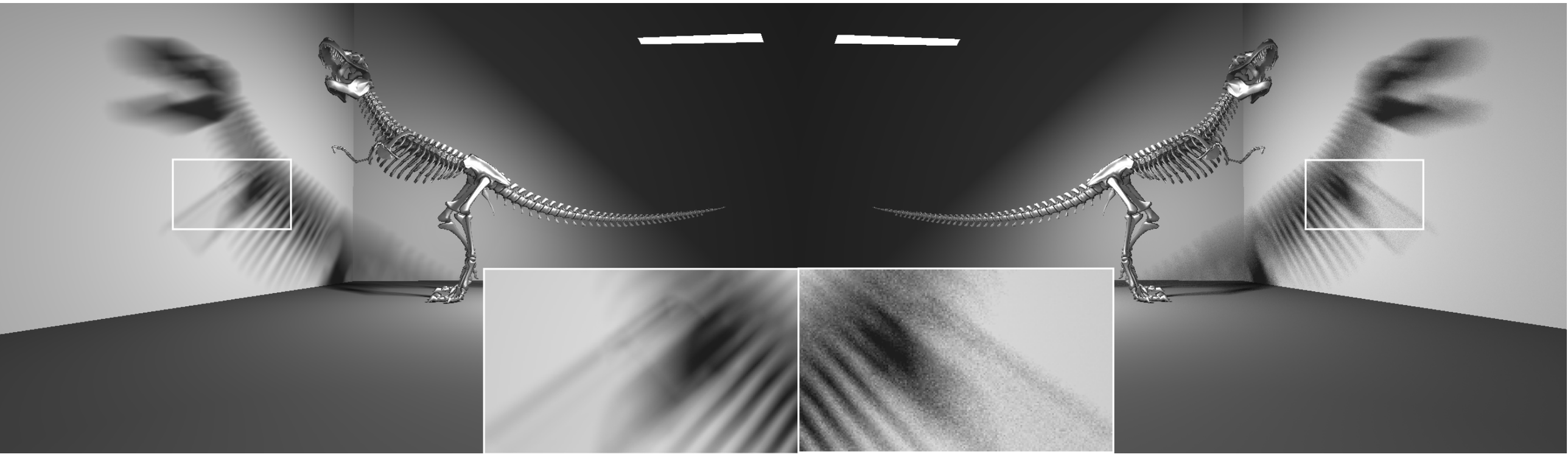








Applications > ombre douces analytiques

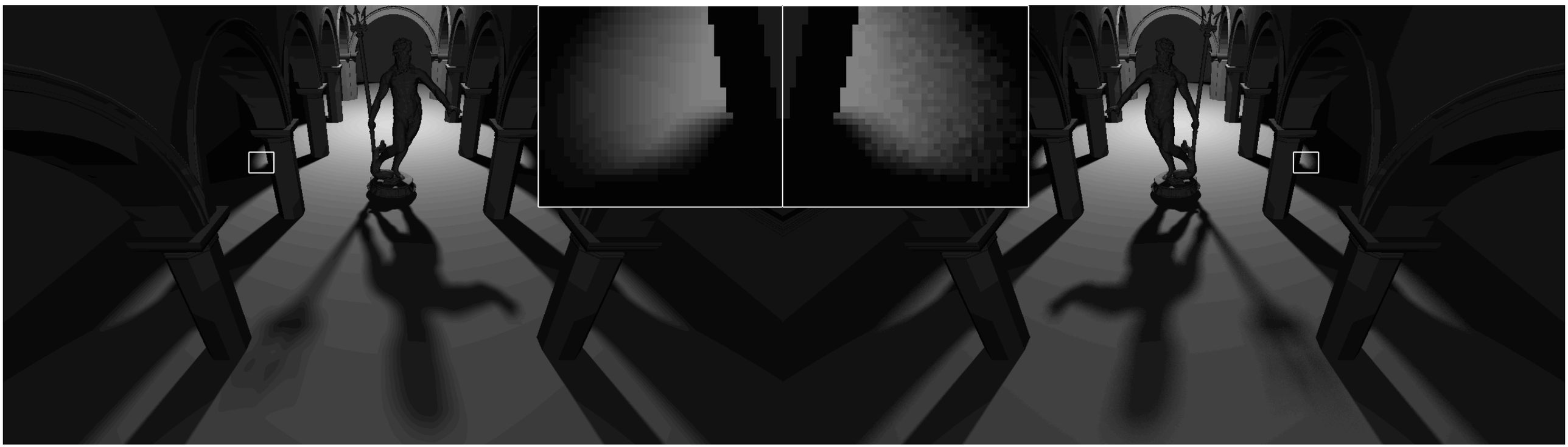


T-Rex (26K triangles)

Time : 6.5s
Memory : 19 MB

Time : 7s
32 samples

Applications > ombre douces analytiques

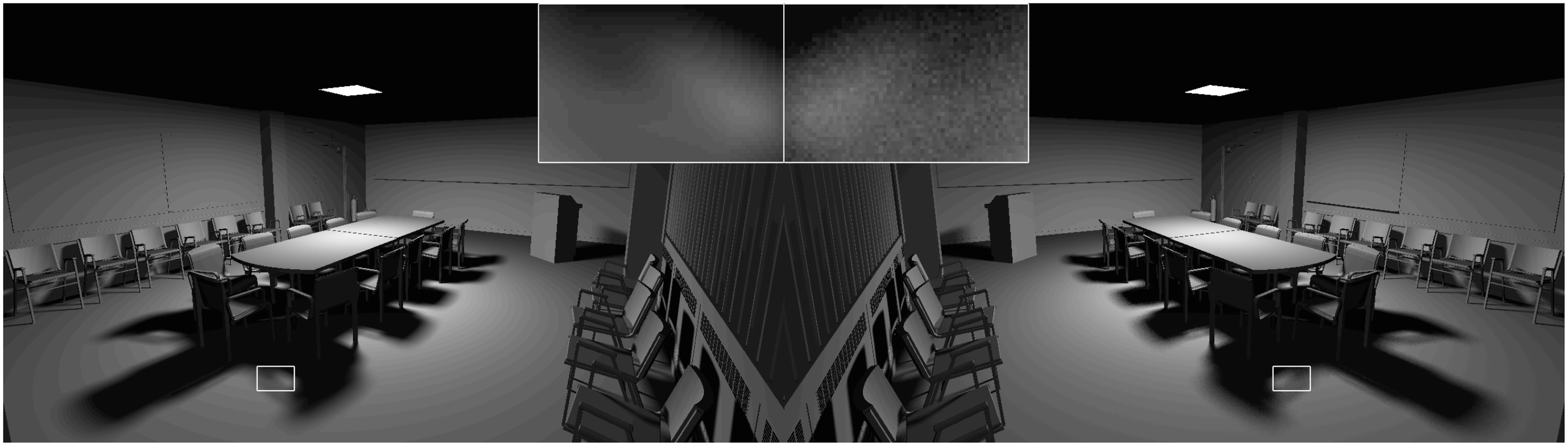


Sponza with Neptune (115K triangles)

Time : 7s
Memory : 16 MB

Time : 7s
32 samples

Applications > ombre douces analytiques

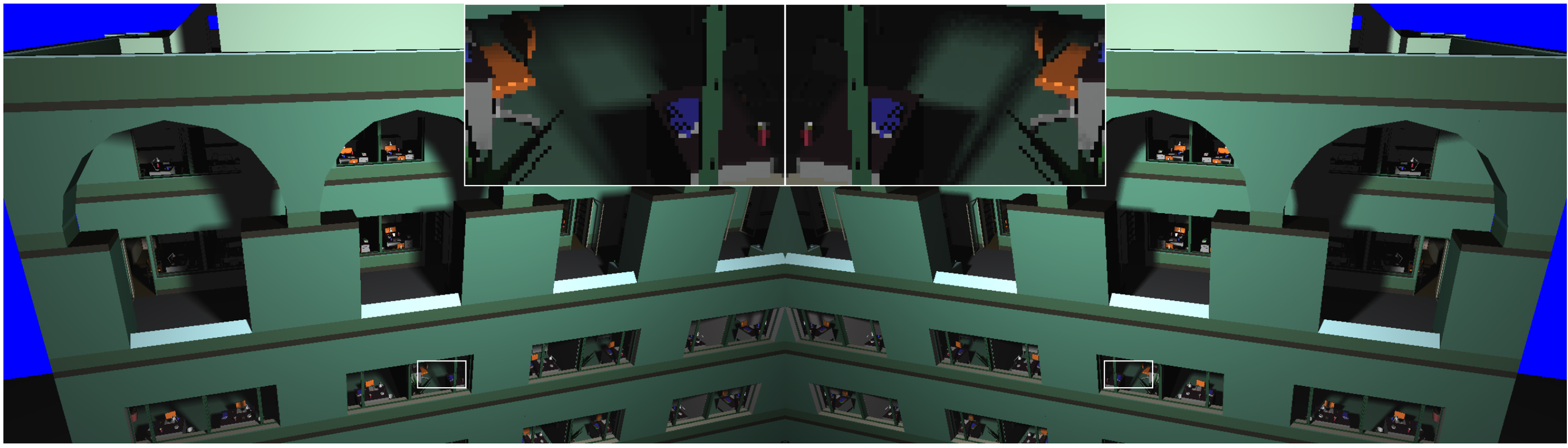


Conference (282K triangles)

Time : 6s
Memory : 20 MB

Time : 6s
32 samples

Applications > ombre douces analytiques

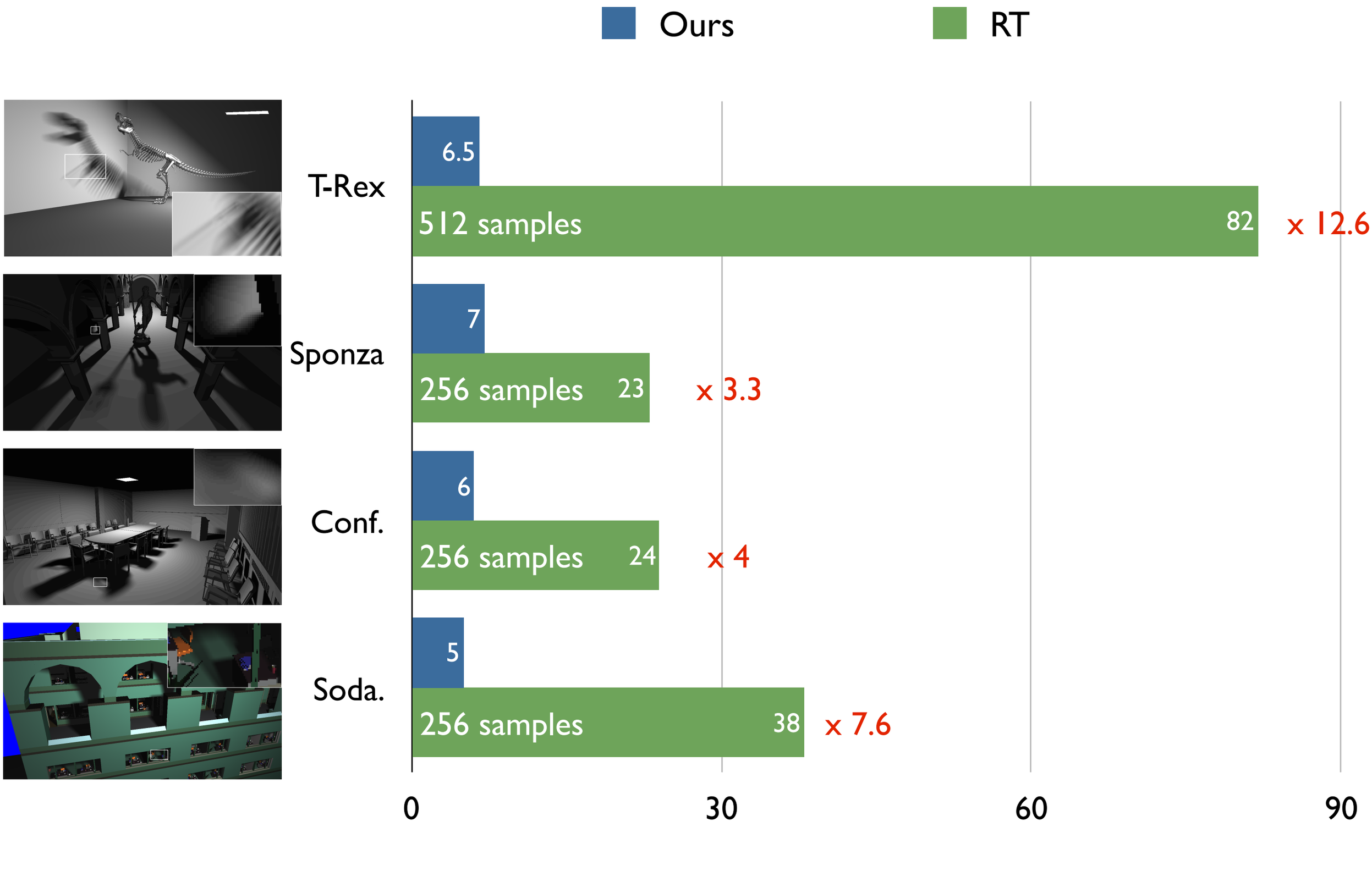


Soda Hall (2147K triangles)

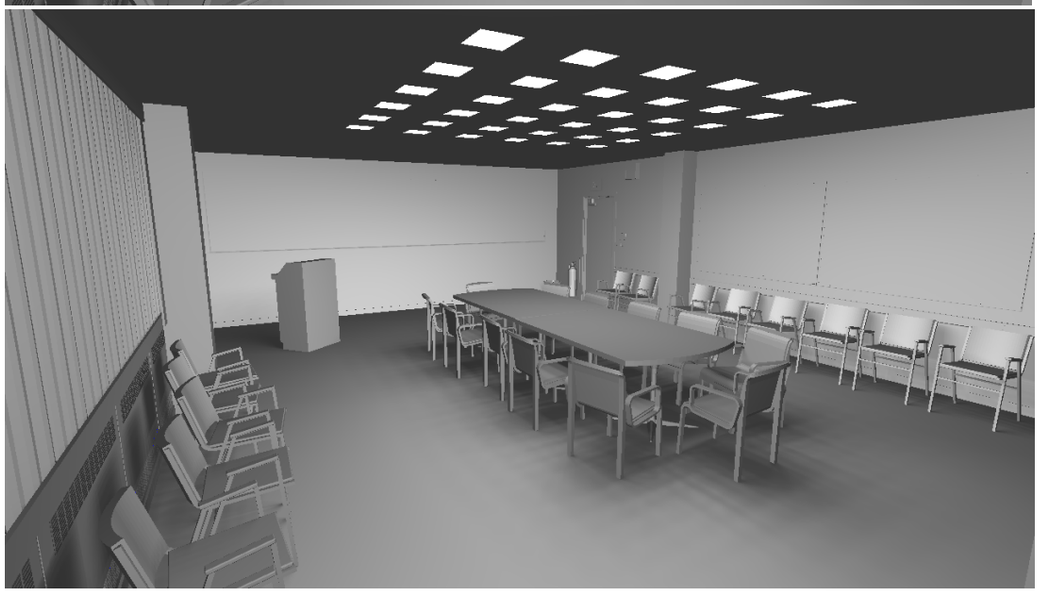
Time : 5s
Memory : 20 MB

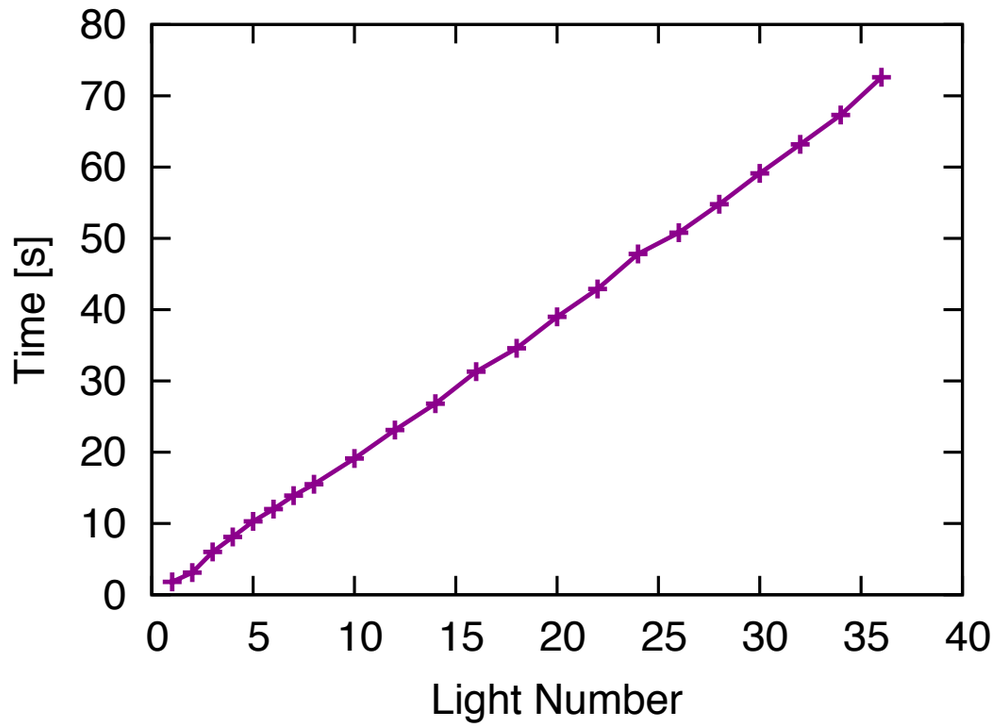
Time : 8s
32 samples

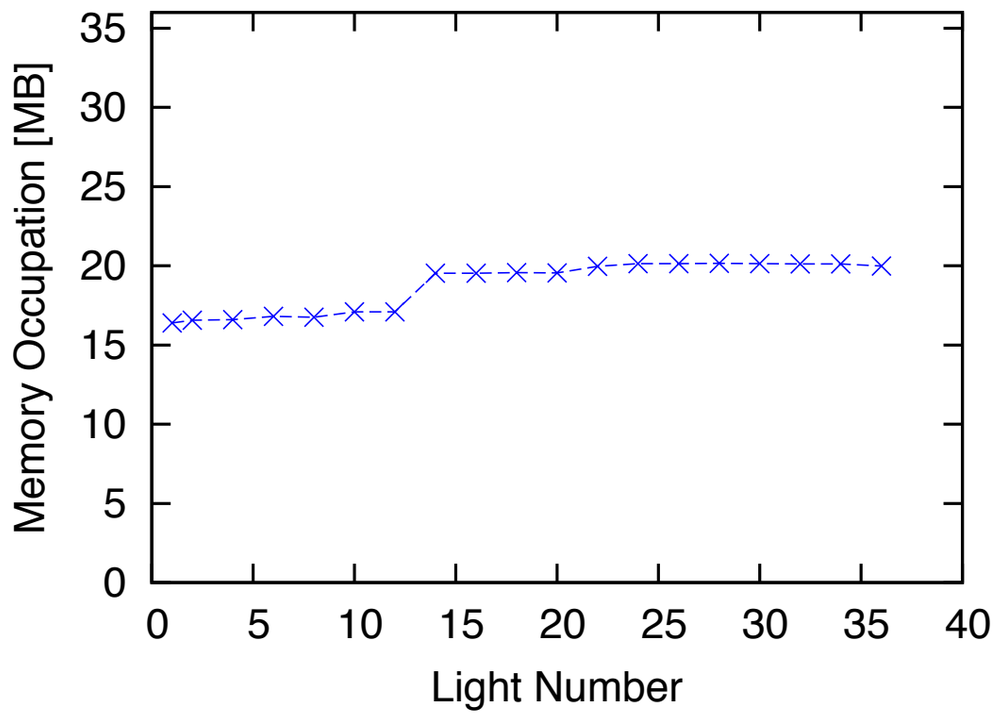
Applications > ombre douces analytiques



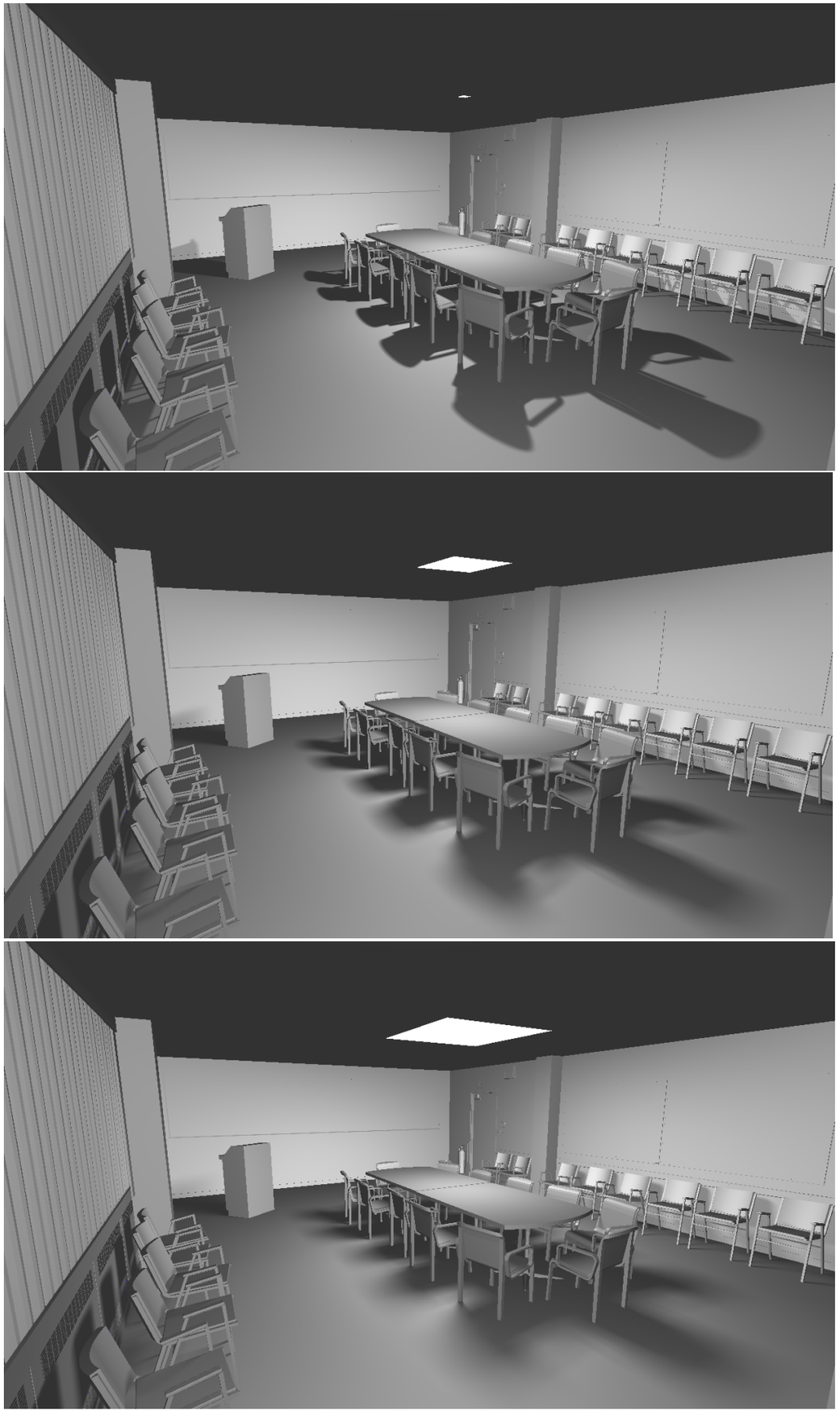
Applications > ombre douces analytiques

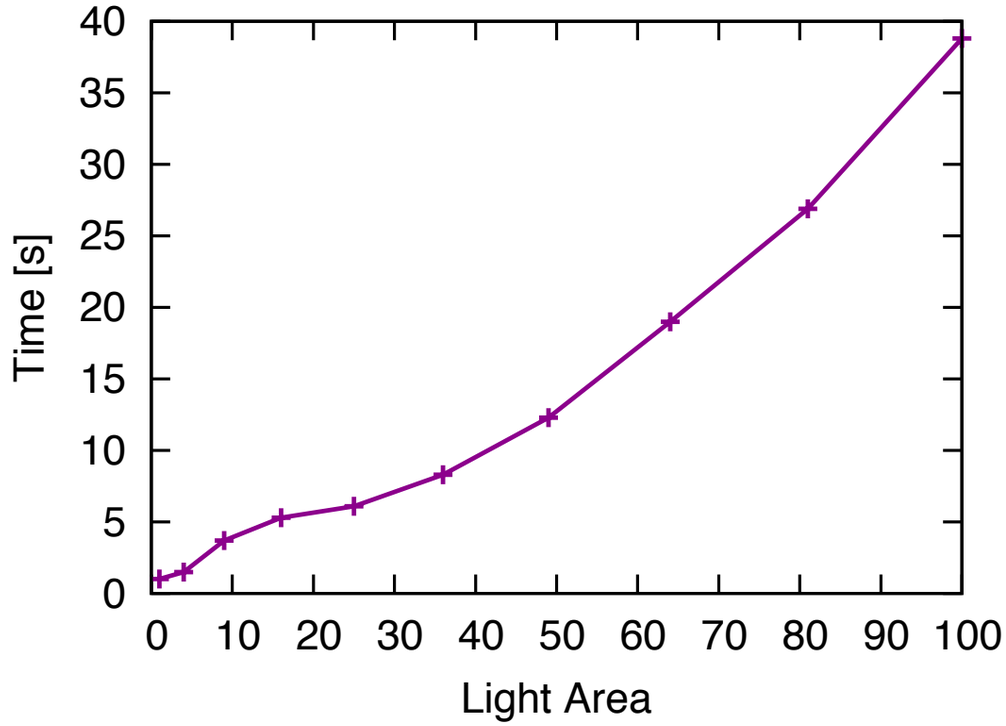


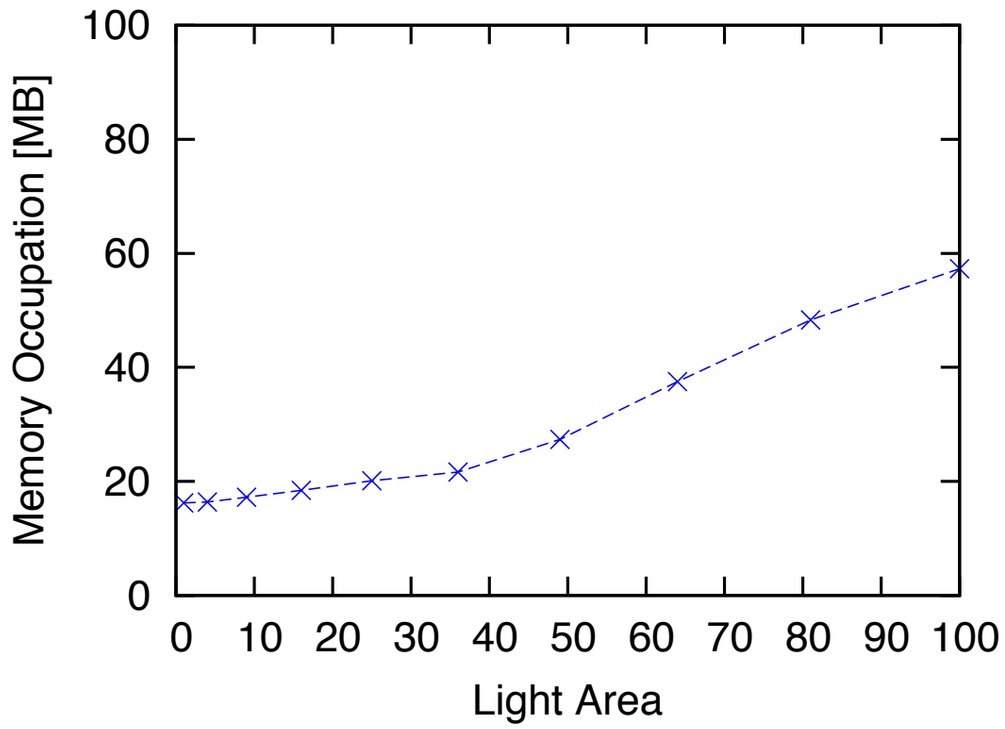




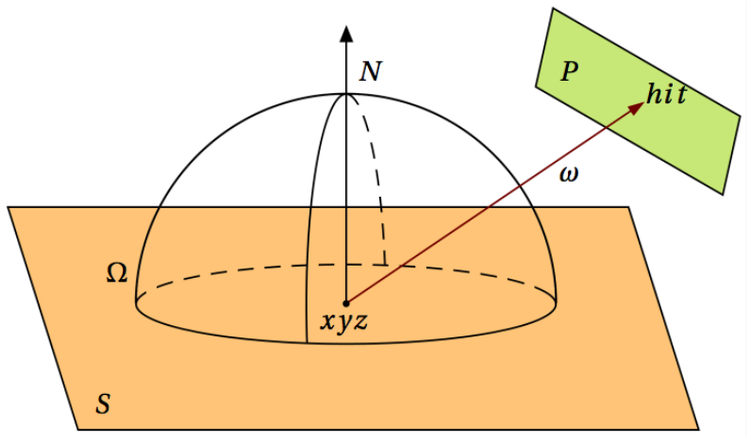
Applications > ombre douces analytiques

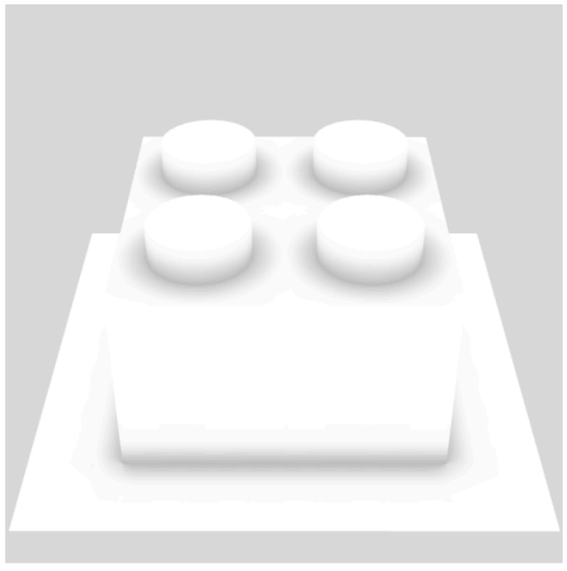


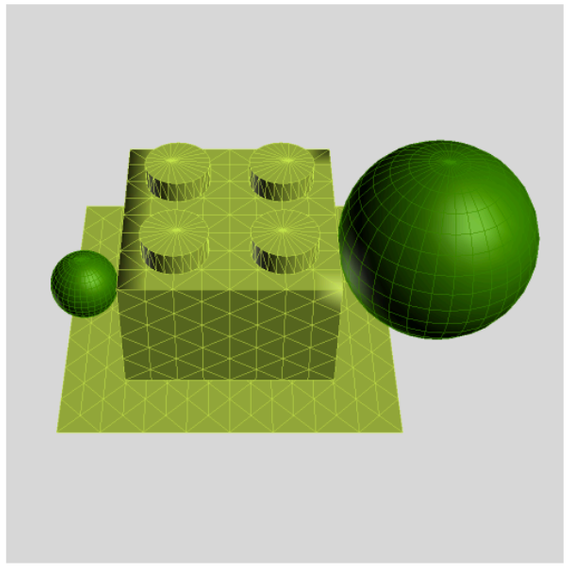


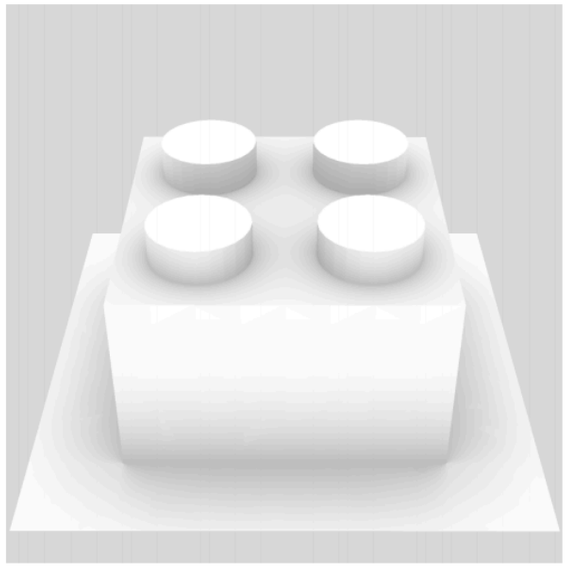


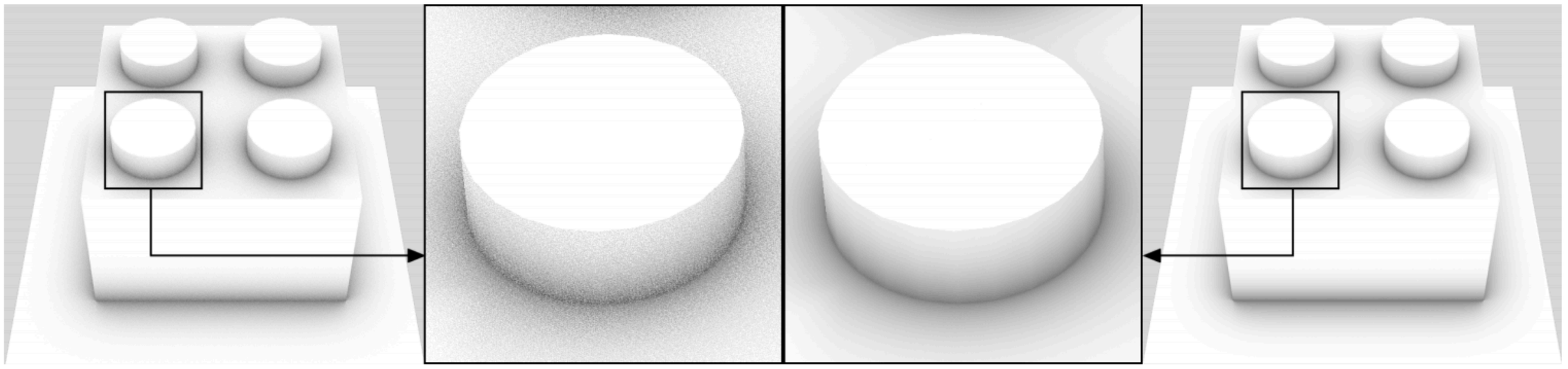
Applications > occlusion ambiante analytique











a) 8 rays / pixel

b) 8 rays / pixel (zoom)

c) 256 rays / pixel (zoom)

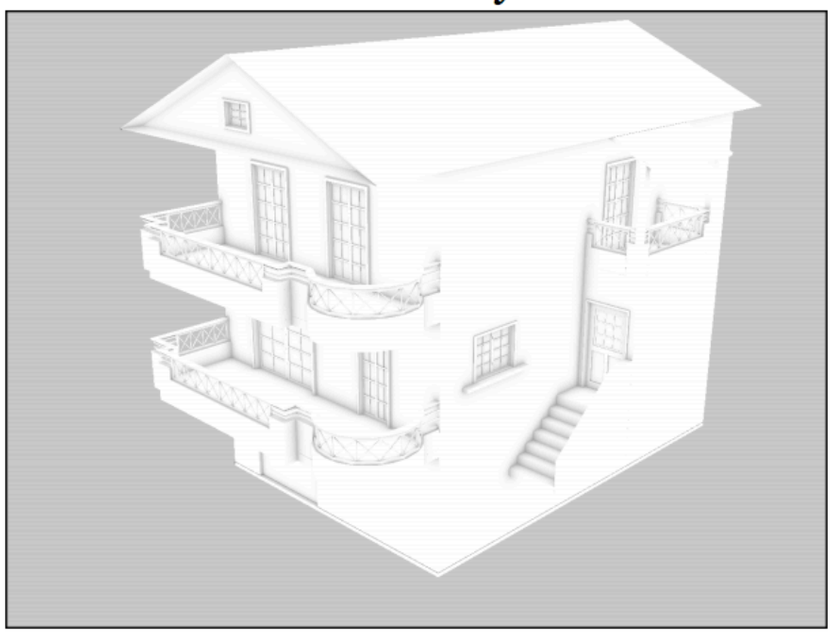
d) 256 rays / pixel

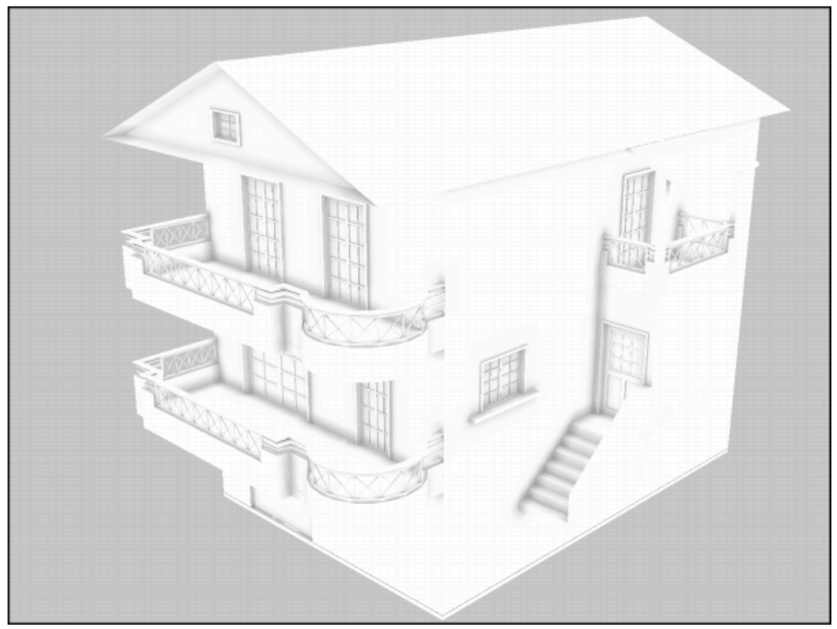
Applications > occlusion ambiante analytique

Mental Ray®

Our method

House
(19K triangles)

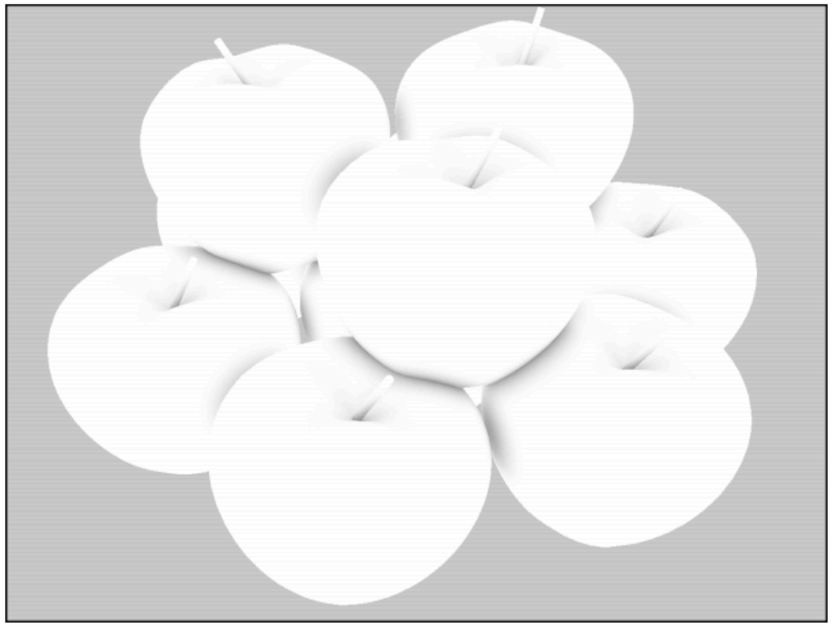


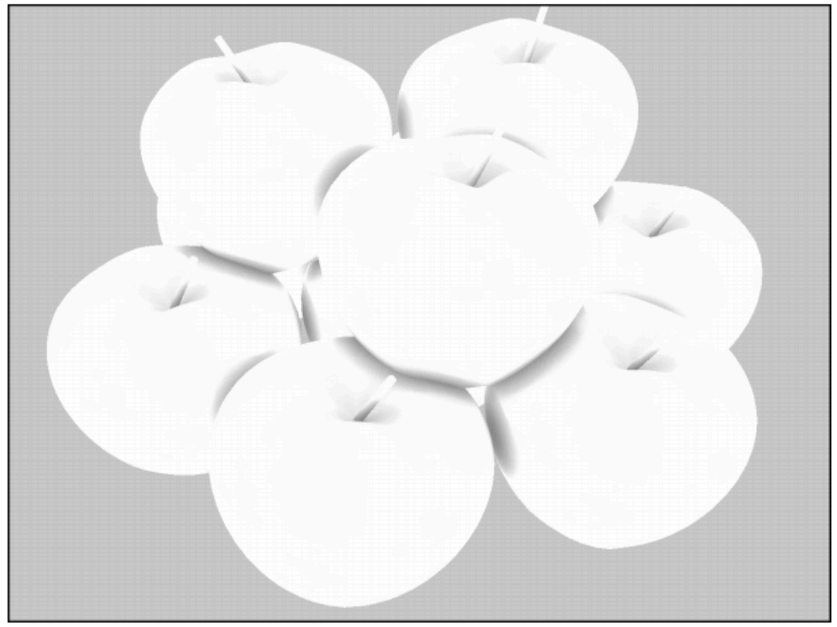


221 seconds

27 seconds

Apples
(32K triangles)





325 seconds

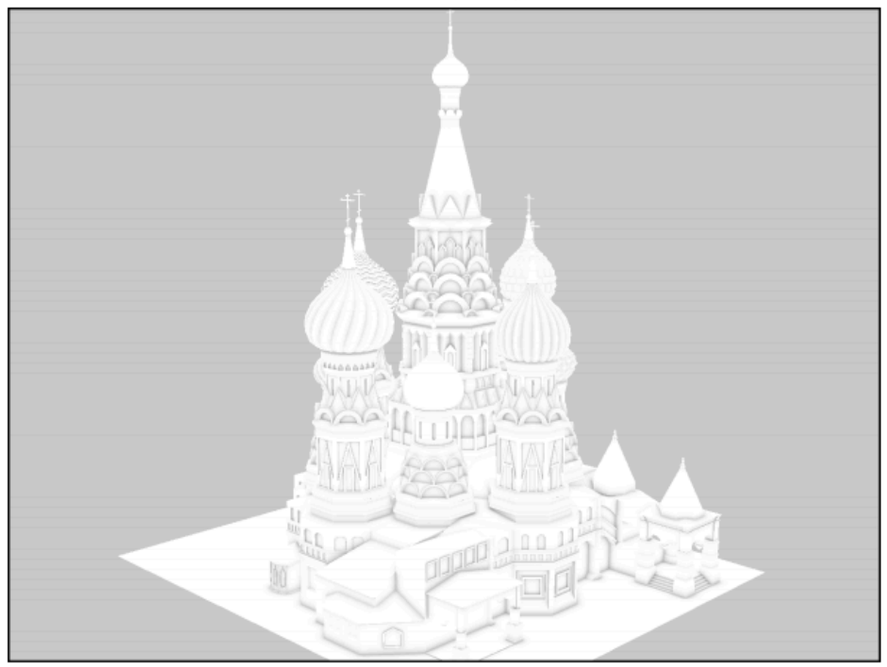
26 seconds

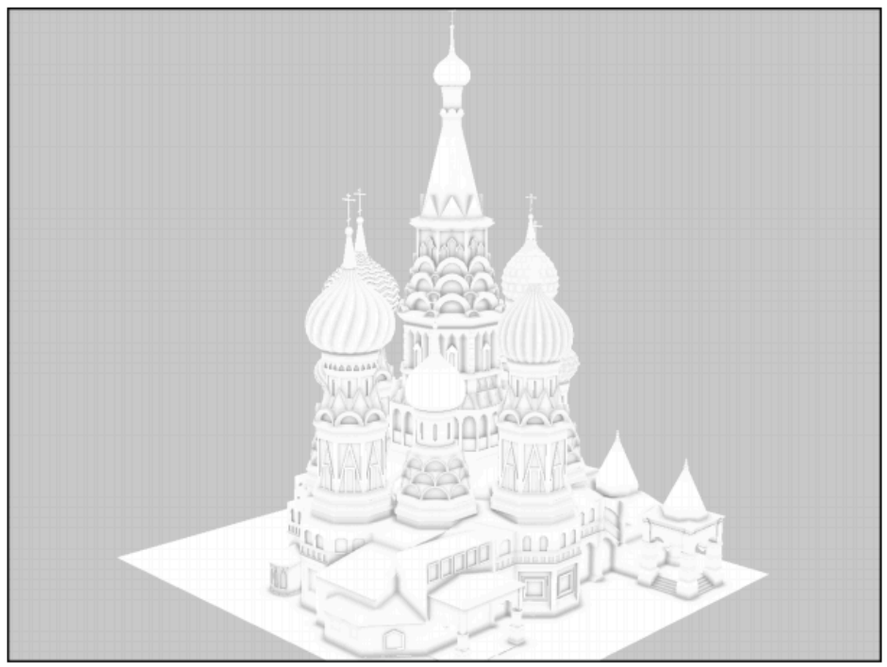
Applications > occlusion ambiante analytique

Mental Ray®

Our method

StBasil
(100K triangles)

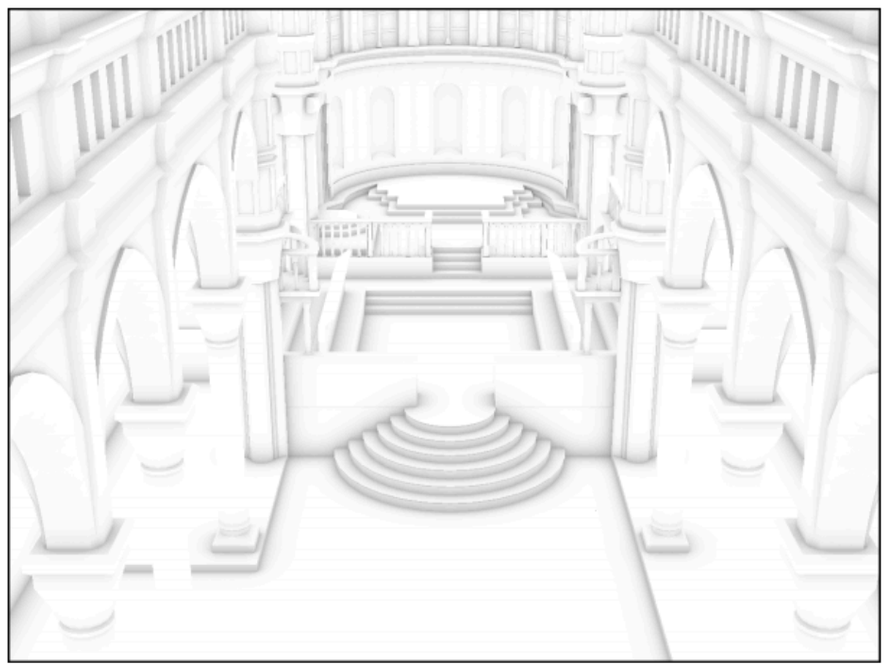


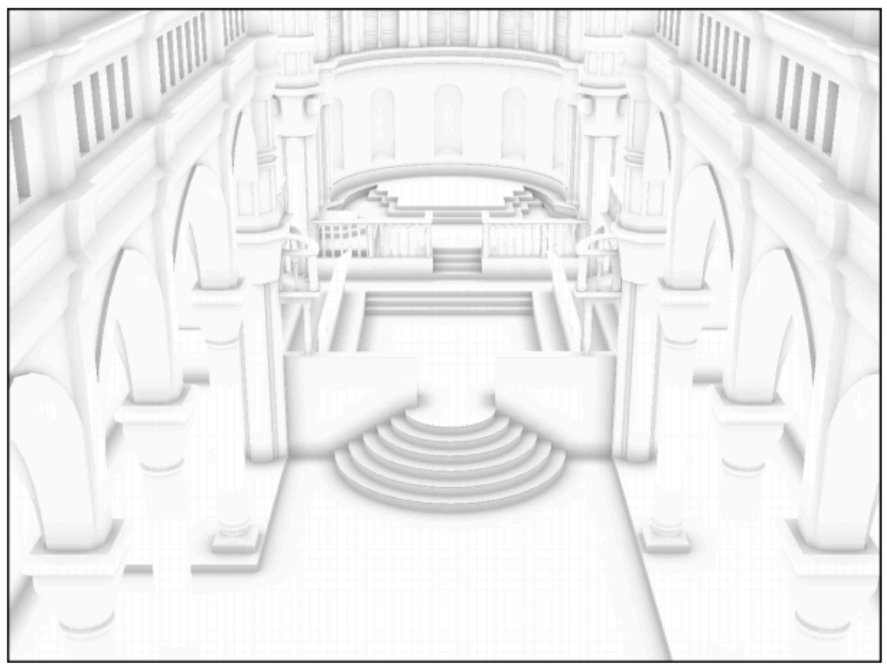


145 seconds

17 seconds

Sibenik
(150K triangles)





547 seconds

49 seconds