Synthèse d'Image et Réalité Virtuelle: Introduction

Ne pas distribuer: certaines images sont protégées par Copyright

ENSICAEN

October 5, 2012

Cette séance

Infos pratiques

Applications

Rapide historique

Pipeline
- Modélisation
- Animations
- Rendu

Thèmes abordés

Cette séance

Infos pratiques

Applications

Rapide historique

Pipeline
- Modélisation
- Animations
- Rendu

Thèmes abordés

Infos Pratiques

- ✓ **Contact**
  - Loïc Simon
  - recherche:
    - Thèse en 2011 en reconstruction procédurale
    - Intérêts: vision par ordinateur, graphics, traitement d'image
  - mail: loic.simon@ensicaen.fr
  - bureau: C108 (GREYC)
- ✓ **Publique visé**
  - ENSICAEN 3A
  - Master Imalang

Fonctionnement

- ✓ **Emploi du temps**
  - ent.unr-runn.fr
  - CM: Jeudi 10h15-12h15, 6 séances
  - TP: Lundi 8h45-10h45, 6 (ou 7?) séances
- ✓ **Documents**
  - foad.ensicaen.fr
- ✓ **Évaluation**
  - TP: rapport, participation
  - Examen de fin de semestre (théorie)

Pré-requis

- ✓ **CM**
  - Outils de maths (géométrie, algèbre linéaire)
  - Notions abordées en OpenGL (maillage, texture, ...)
- ✓ **TP**
  - C/C++
  - OpenGL

Contenu du module

- ✓ **CM** (6 séances)
  - Rendu réaliste
  - Théorique
- ✓ **TP** (7 séances)
  - Continuation 2A
  - OpenGL
  - Ray tracing??

Références

- ✓ **CM**
  - books?
  - inst.eecs.berkeley.edu/ cs184/fa12/index.html

- ✓ **TP**

  Doc officielle
  - www.opengl.org/sdk/
  - red book, orange book

  Tutoriaux
  - www.lighthouse3d.com/tutorials/glsl-core-tutorial
  - duriansoftware.com/joe/An-intro-to-modern-OpenGL.-Table-of-Contents.html
  - www.opengl-tutorial.org/
  - prideout.net/blog/?p=48

Cette séance

Infos pratiques

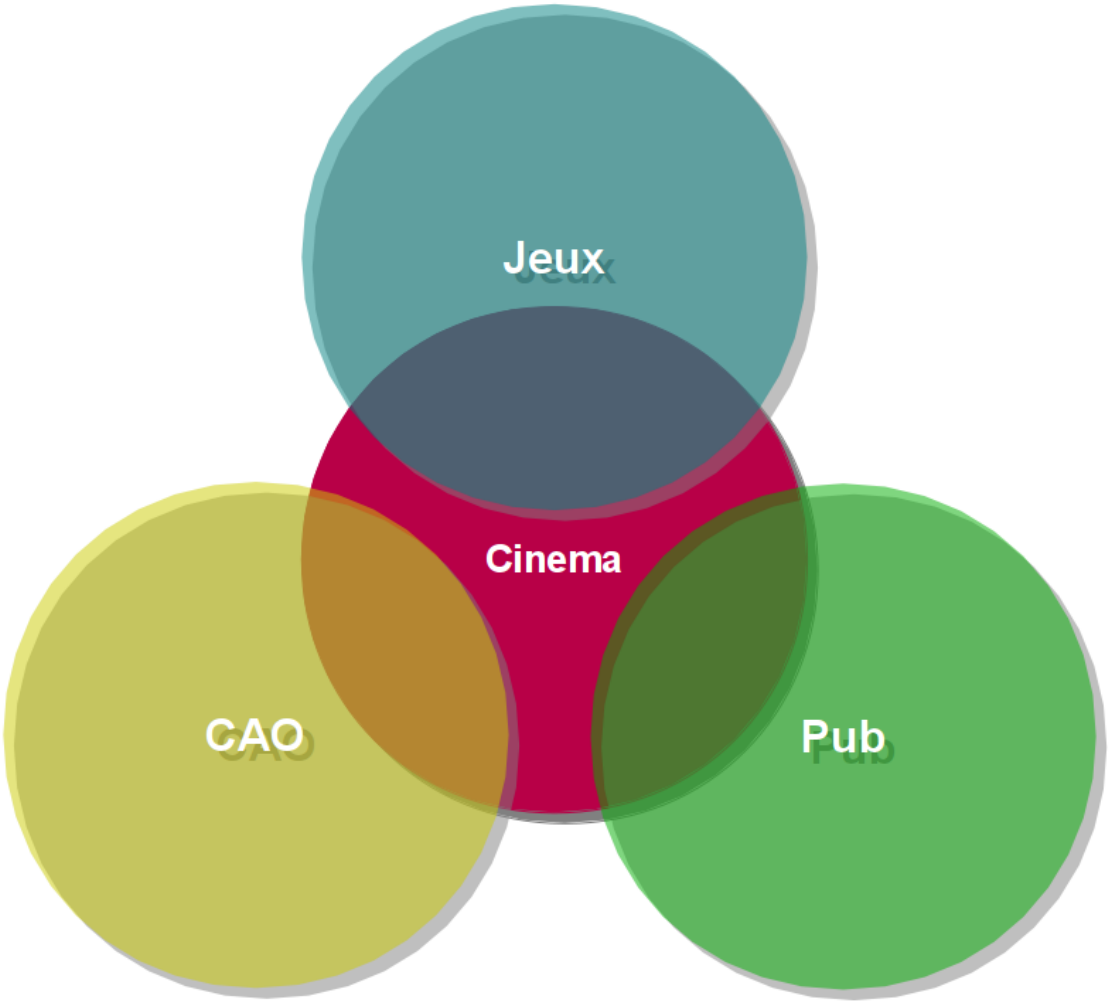
Applications

Rapide historique

Pipeline
- Modélisation
- Animations
- Rendu

Thèmes abordés

Applications et spécificités



Applications et spécificités (films)

Figure: Brave, Pixar 2012

Spécificités

- ✓ Réalisme
  - rendu
  - animation
- ✗ temps réel

Applications et spécificités (films)

Spécificités

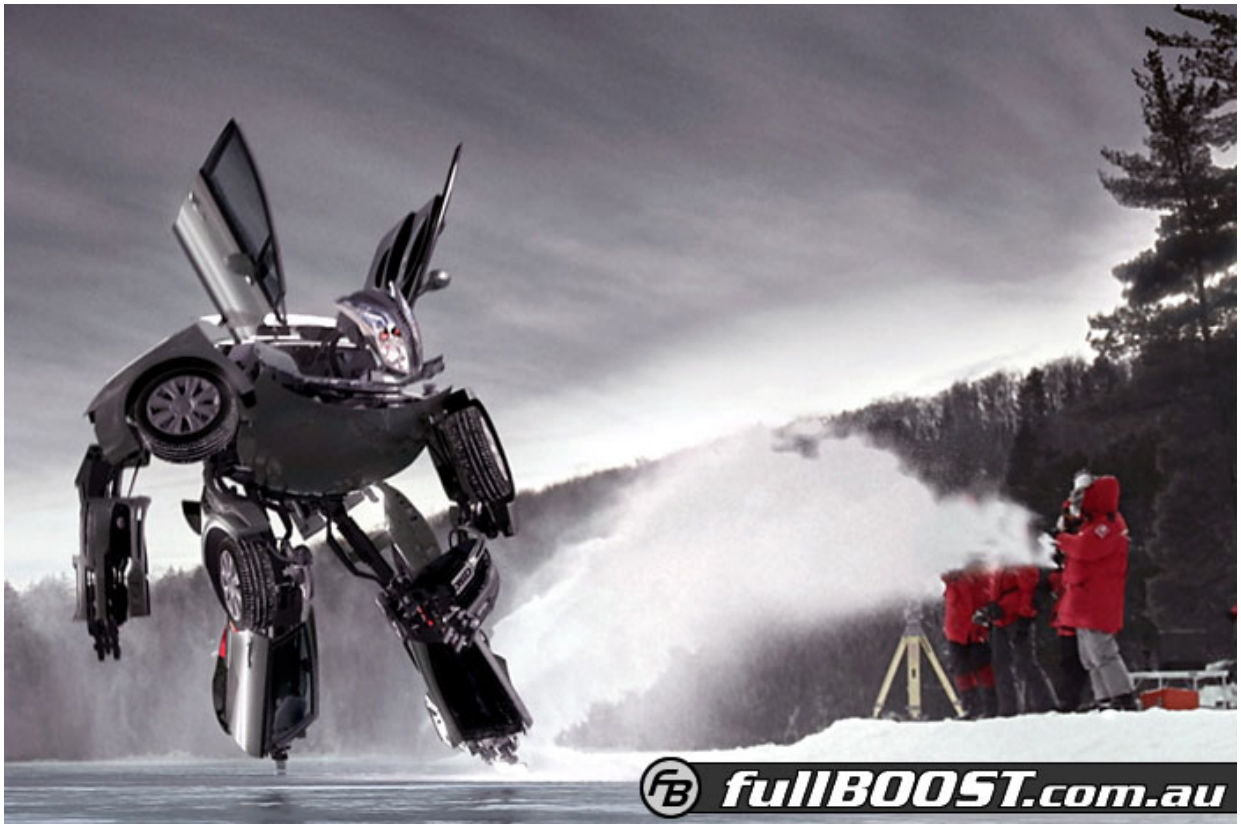
- ✓ Réalisme
  - rendu
  - animation
- ✗ temps réel

Applications et spécificités (pub)

Spécificités

- ✓ Réalisme
  - rendu
  - animation
- ✗ temps réel

Applications et spécificités (pub)



Spécificités

- ✓ Réalisme
  - rendu
  - animation
- ✗ temps réel

Applications et spécificités (jeux)

Figure: Halo 3, Bungie 2007

Spécificités

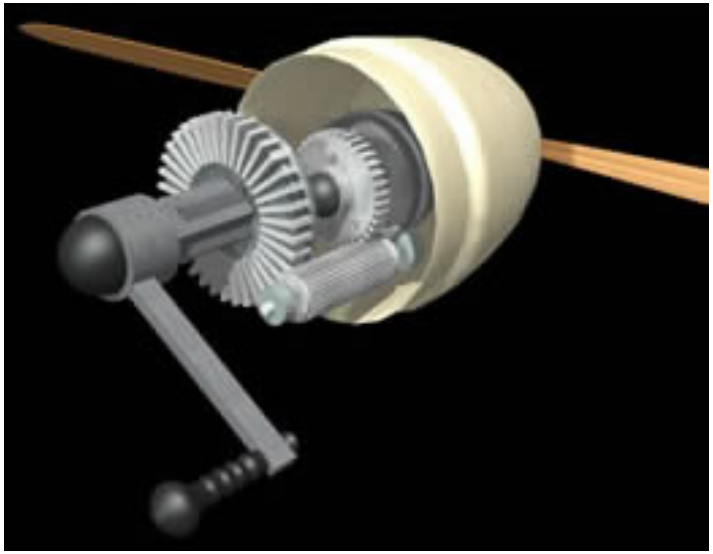
- ✓ Réalisme
  - rendu
  - animations
  - IA
- ✓ immersion
- ✓ temps réel

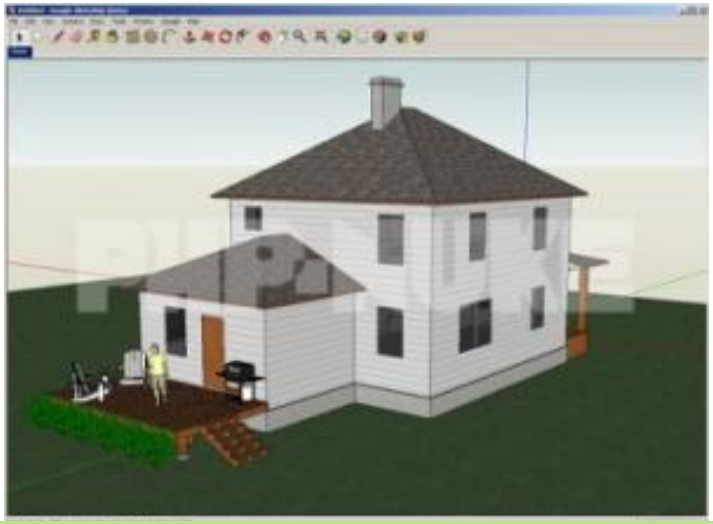
Applications et spécificités (réalité virtuelle & augmentée)

Spécificités

- ✓ Réalisme
  - rendu
  - fusion (réel-virtuel)
- ✓ immersion
- ✓ temps réel

Applications et spécificités (CAO)

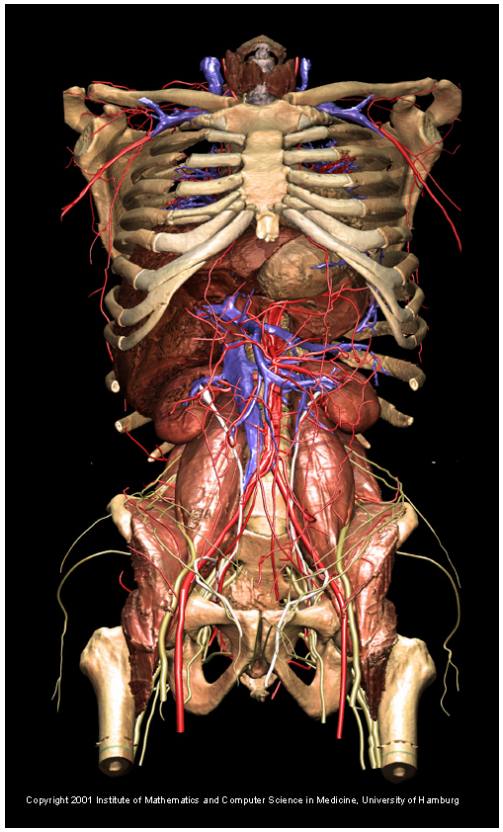




Spécificités

- ✓ Réalisme
  - Comportement physique
- ✓ temps réel

Applications et spécificités (science)



Spécificités

- ✓ Réalisme
  - anatomique
  -

Cette séance

Infos pratiques

Applications

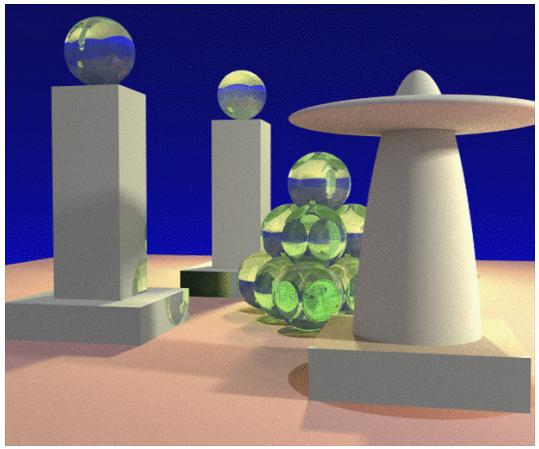
Rapide historique

Pipeline
- Modélisation
- Animations
- Rendu

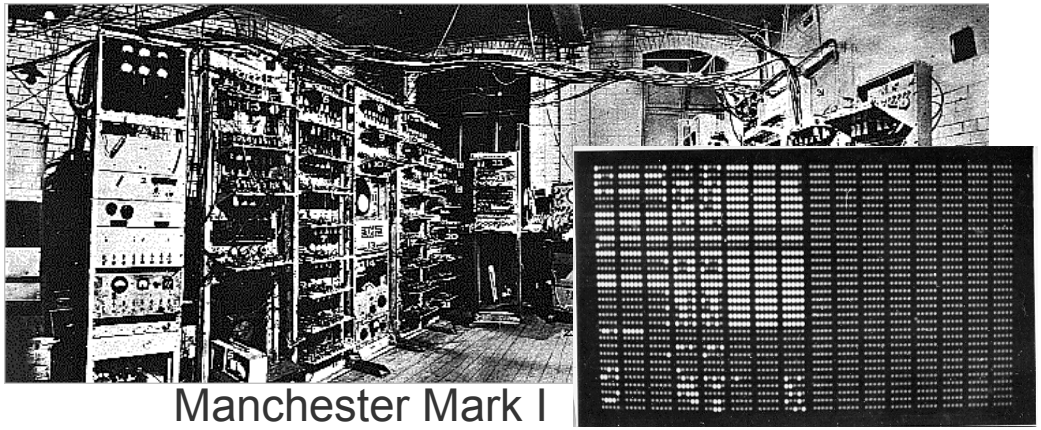
Thèmes abordés

Historique de la Synthèse d'Images (CGI)

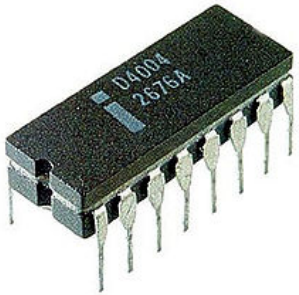
- ✓ en.wikipedia.org/wiki/History_of_computer_animation
- ✓ www.youtube.com/watch?v=LzZwiLUVaKg (10 min)



Historique de la Synthèse d'Images (Hardware)



Manchester Mark I



Microprocessor (71)

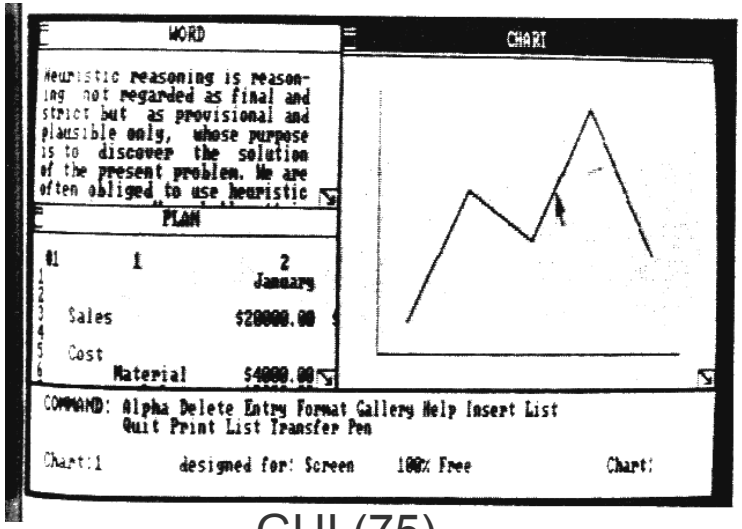
Xerox Alto (72)

Framebuffer (74)

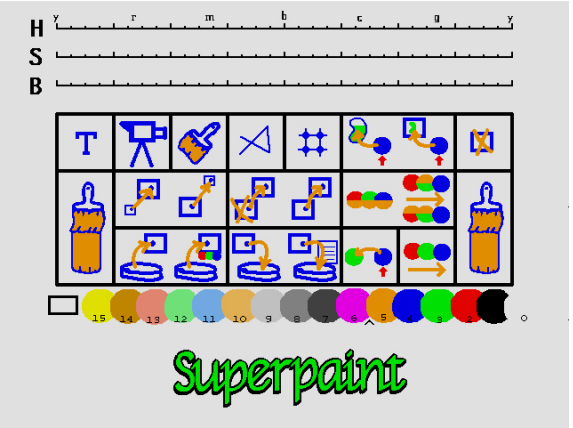
Apple II (77)

Historique de la Synthèse d'Images (Software)

SketchPad (63)



GUI (75)



Cette séance

Infos pratiques

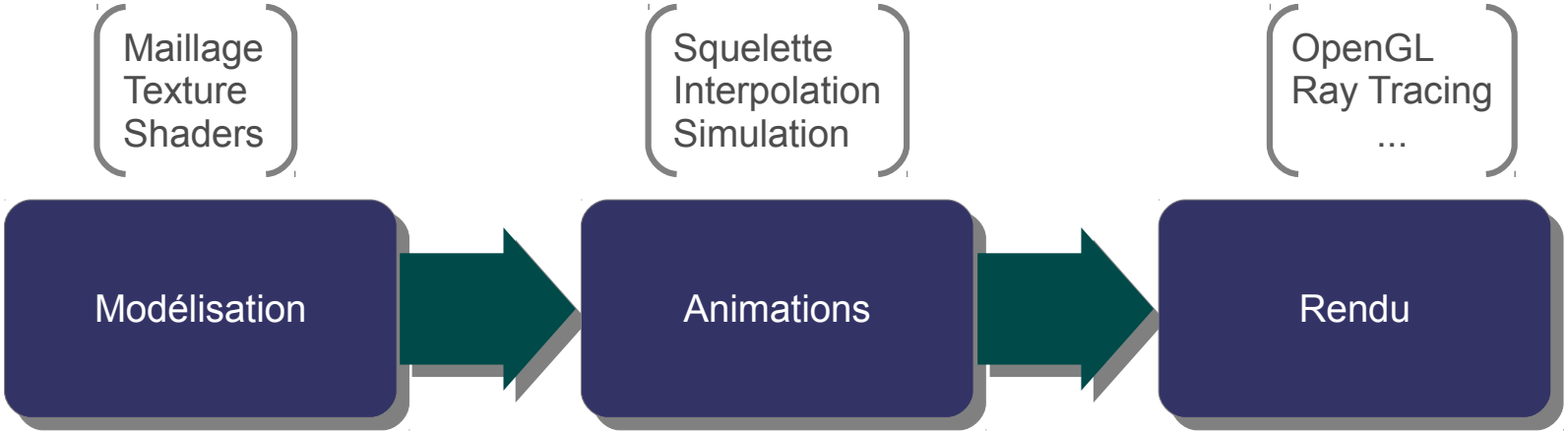
Applications

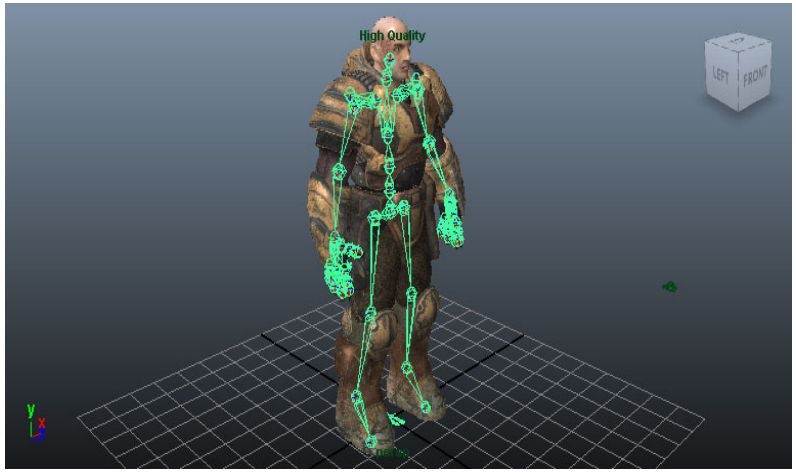
Rapide historique

Pipeline
- Modélisation
- Animations
- Rendu

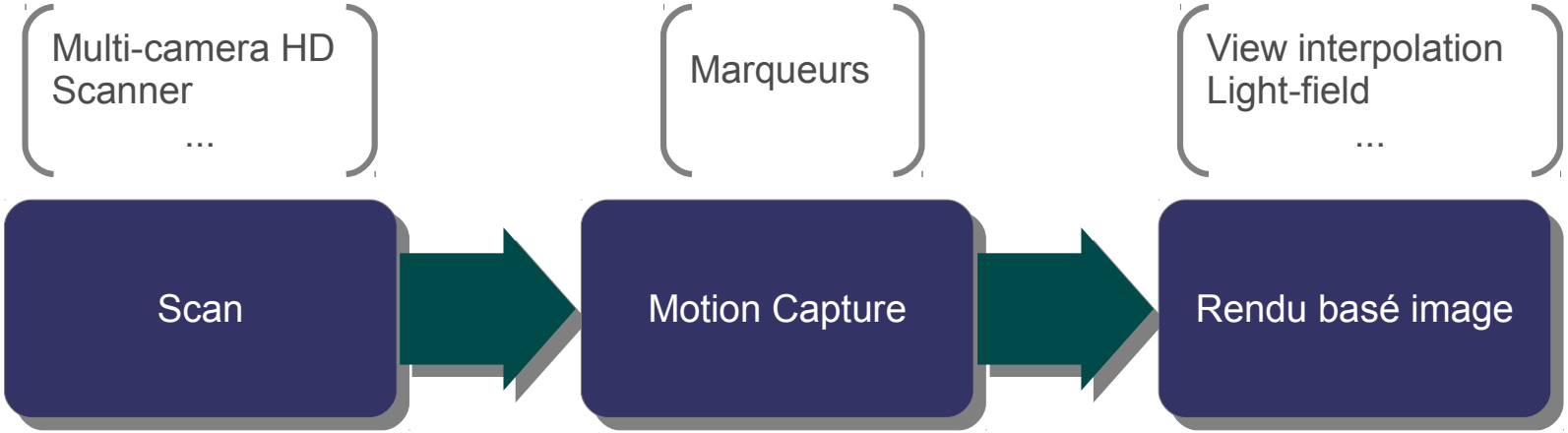
Thèmes abordés

Pipeline (classique)

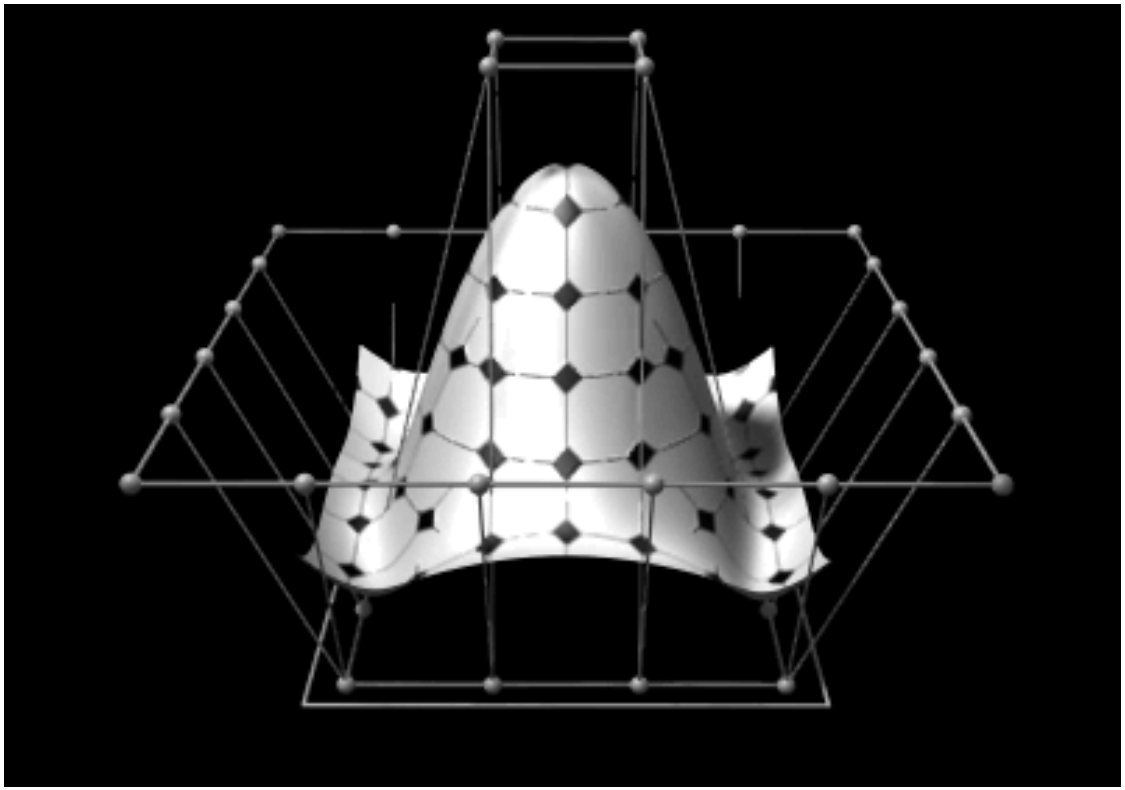




Pipeline (motion-scan)

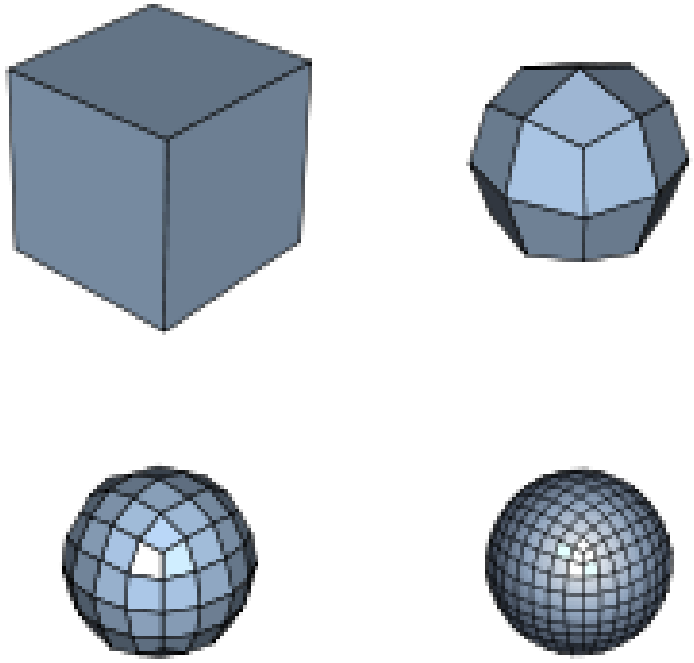


Modélisation: rapide historique



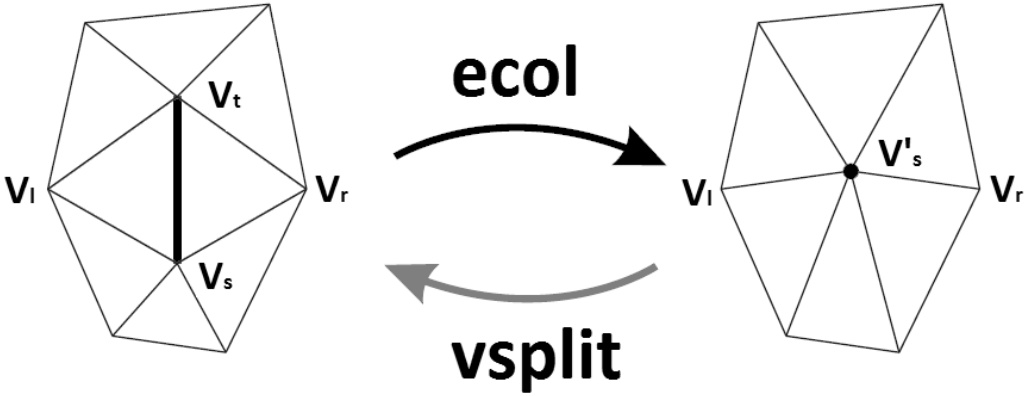
- ✓ maillage
- ✓ B-splines, Bézier curves, NURBS [Bézier 50]
- ✓ subdivision surfaces: [Catmull-Clark 78]
- ✓ progressive meshes: [Hoppe 96]
- ✓ ...

Modélisation: rapide historique



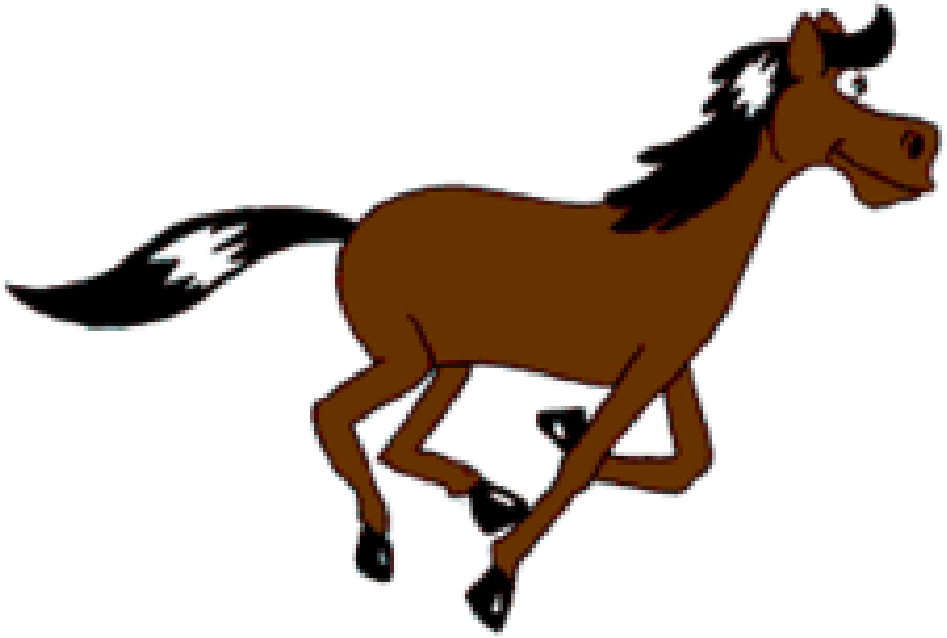
- ✓ maillage
- ✓ B-splines, Bézier curves, NURBS [Bézier 50]
- ✓ subdivision surfaces: [Catmull-Clark 78]
- ✓ progressive meshes: [Hoppe 96]
- ✓ ...

Modélisation: rapide historique



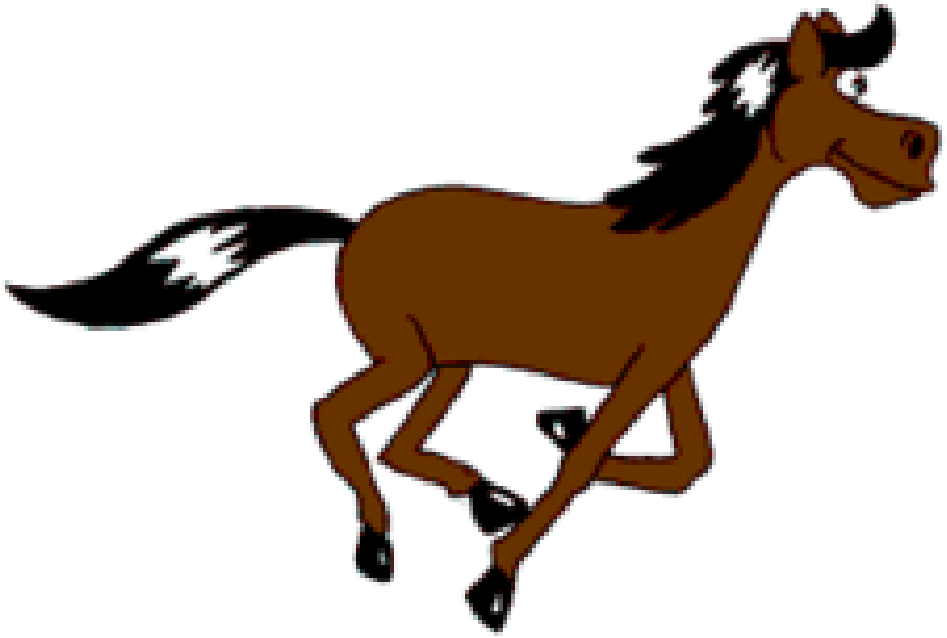
- ✓ maillage
- ✓ B-splines, Bézier curves, NURBS [Bézier 50]
- ✓ subdivision surfaces: [Catmull-Clark 78]
- ✓ progressive meshes: [Hoppe 96]
- ✓ ...

Animations



- ✓ Succession d'images
- ✓ Dessinateurs
- ✓ Morphing (frame interpolation)
- ✓ 3D animation
  - rigides
  - articulés
  - vêtements
  - liquide
  - ...

Animations



- ✓ Succession d'images
- ✓ Dessinateurs
- ✓ Morphing (frame interpolation)
- ✓ 3D animation
  - rigides
  - articulés
  - vêtements
  - liquide
  - ...

Animations

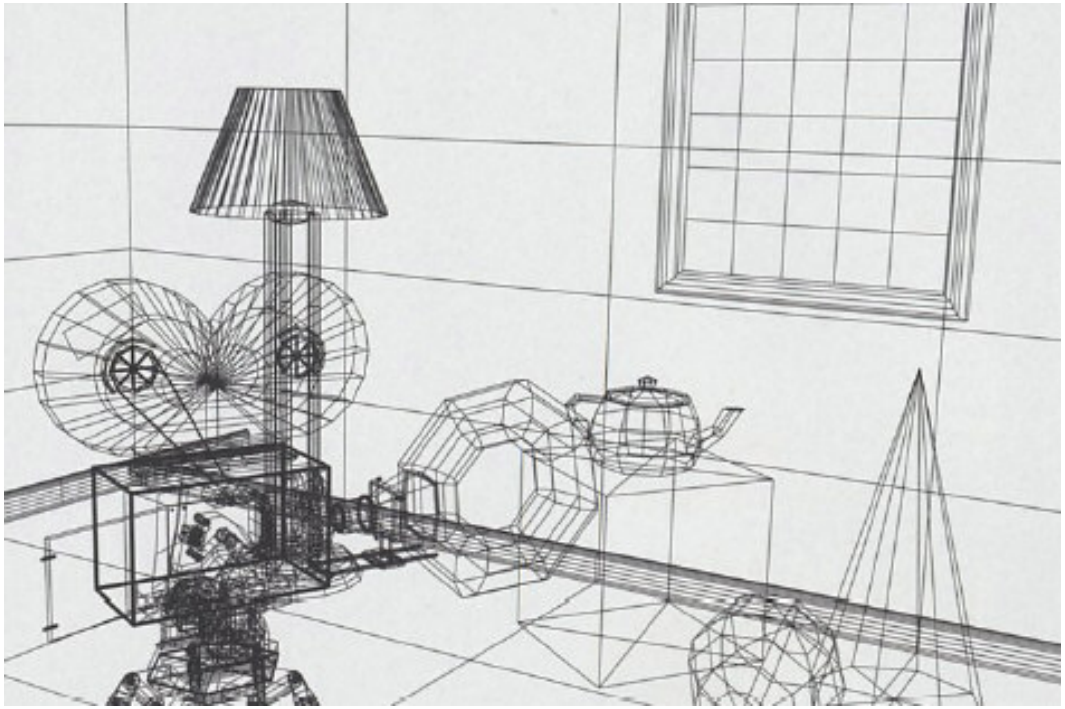
- ✓ Succession d'images
- ✓ Dessinateurs
- ✓ Morphing (frame interpolation)
- ✓ 3D animation
  - rigides
  - articulés
  - vêtements
  - liquide
  - ...

Animations

- ✓ Succession d'images
- ✓ Dessinateurs
- ✓ Morphing (frame interpolation)
- ✓ 3D animation
  - rigides
  - articulés
  - vêtements
  - liquide
  - ...

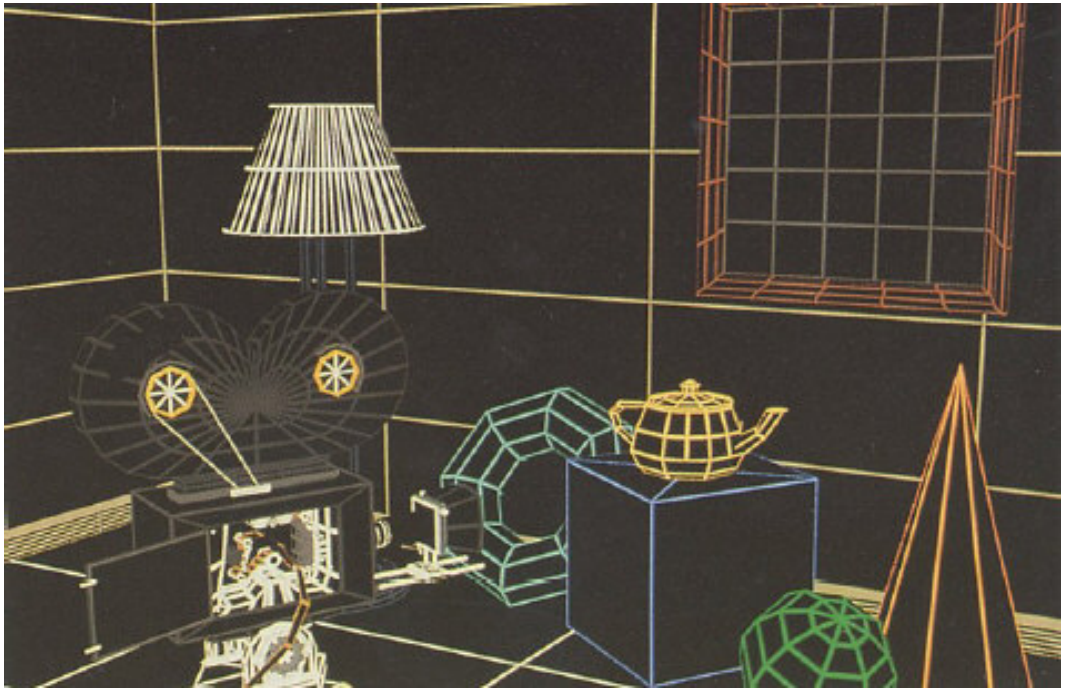
Rendu: Rôle d'une théière

Rendu: visibilité (60's)



- ✓ hidden-line algorithms: [Roberts 63], [Appel 67]
- ✓ hidden-surface: [Warnock 69], [Watkins 70]
- ✓ "visibilité = sorting": [Sutherland 74]
- ✓ Z-buffer: [Catmull 74]

Rendu: visibilité (60's)



- ✓ hidden-line algorithms: [Roberts 63], [Appel 67]
- ✓ hidden-surface: [Warnock 69], [Watkins 70]
- ✓ "visibilité = sorting": [Sutherland 74]
- ✓ Z-buffer: [Catmull 74]

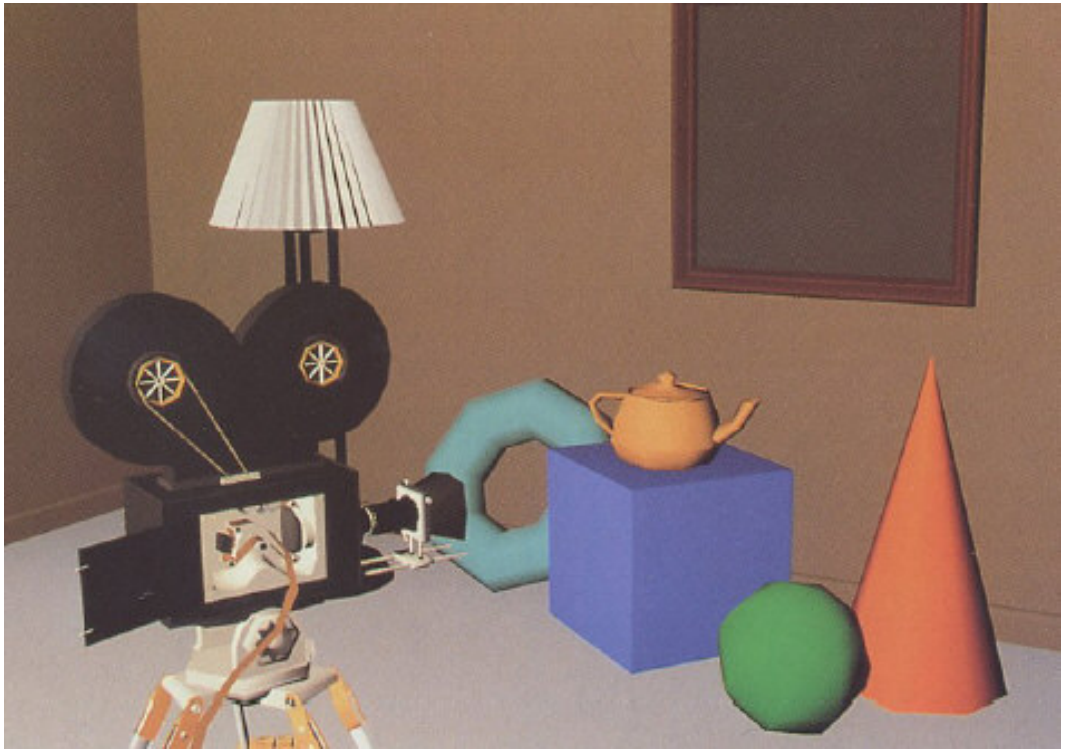
Rendu: visibilité (60's)

- ✓ hidden-line algorithms: [Roberts 63], [Appel 67]
- ✓ hidden-surface: [Warnock 69], [Watkins 70]
- ✓ "visibilité = sorting": [Sutherland 74]
- ✓ Z-buffer: [Catmull 74]

Rendu: shading (70's)

- ✓ illumination diffuse: [Gouraud 71]
- ✓ illumination spéculaire: [Phong 74]
- ✓ texture: [Blinn 74]

Rendu: shading (70's)

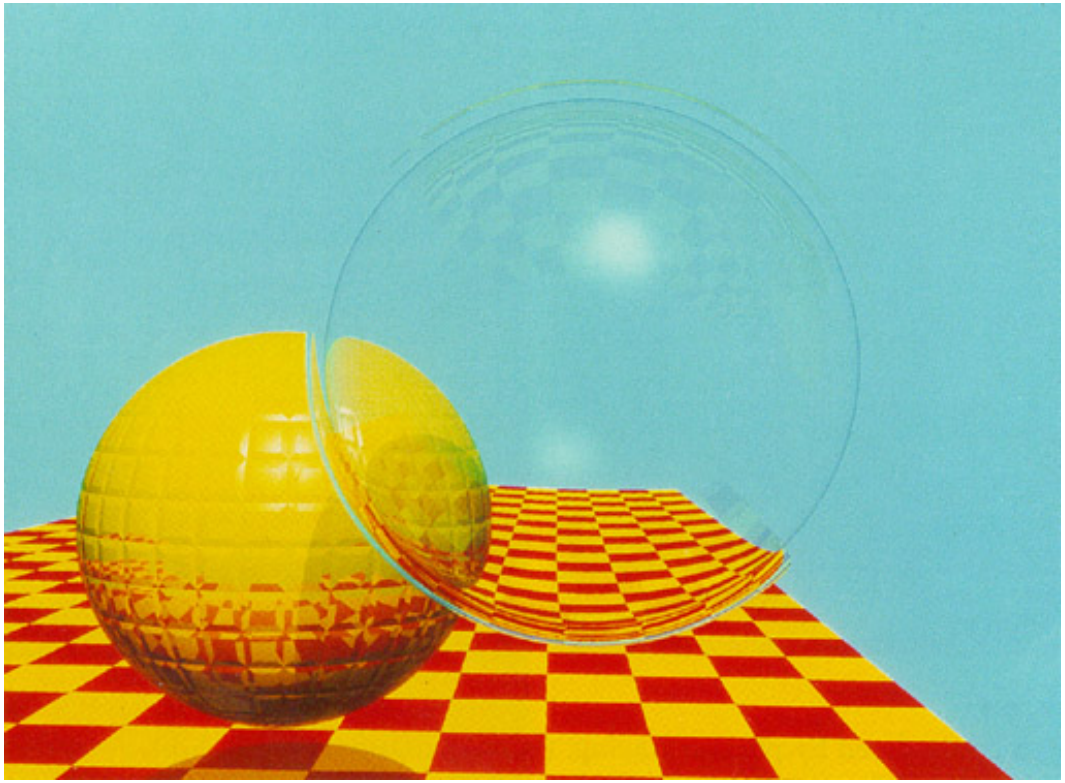


- ✓ illumination diffuse: [Gouraud 71]
- ✓ illumination spéculaire: [Phong 74]
- ✓ texture: [Blinn 74]

Rendu: shading (70's)

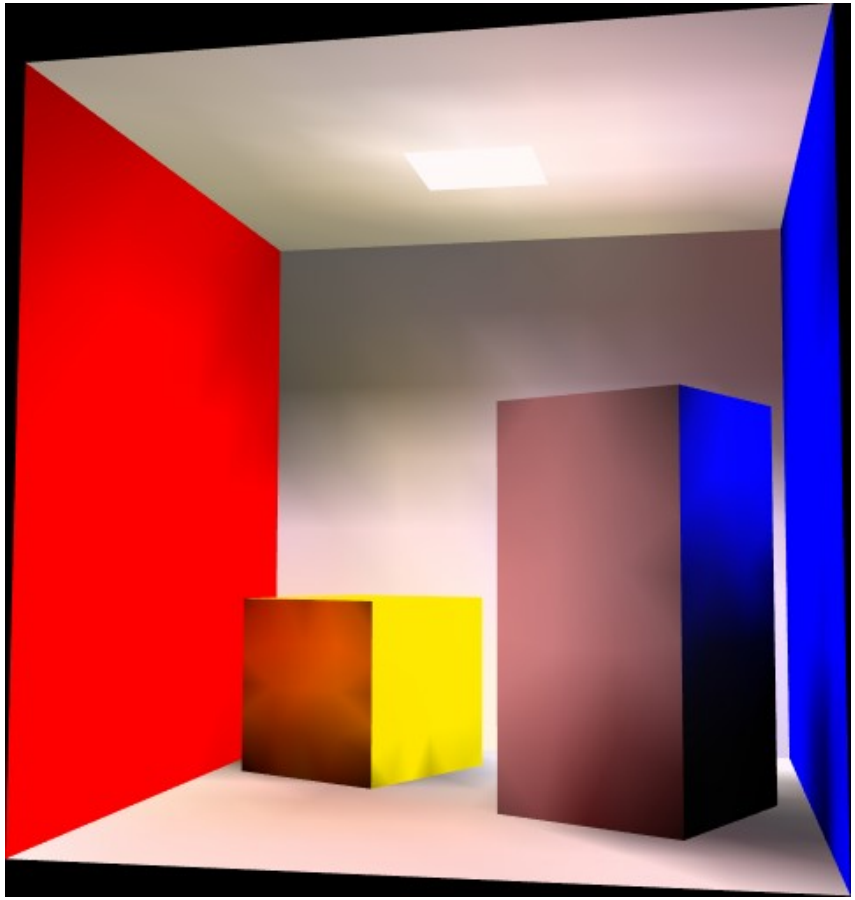
- ✓ illumination diffuse: [Gouraud 71]
- ✓ illumination spéculaire: [Phong 74]
- ✓ texture: [Blinn 74]

Rendu: illumination globale (80's -)



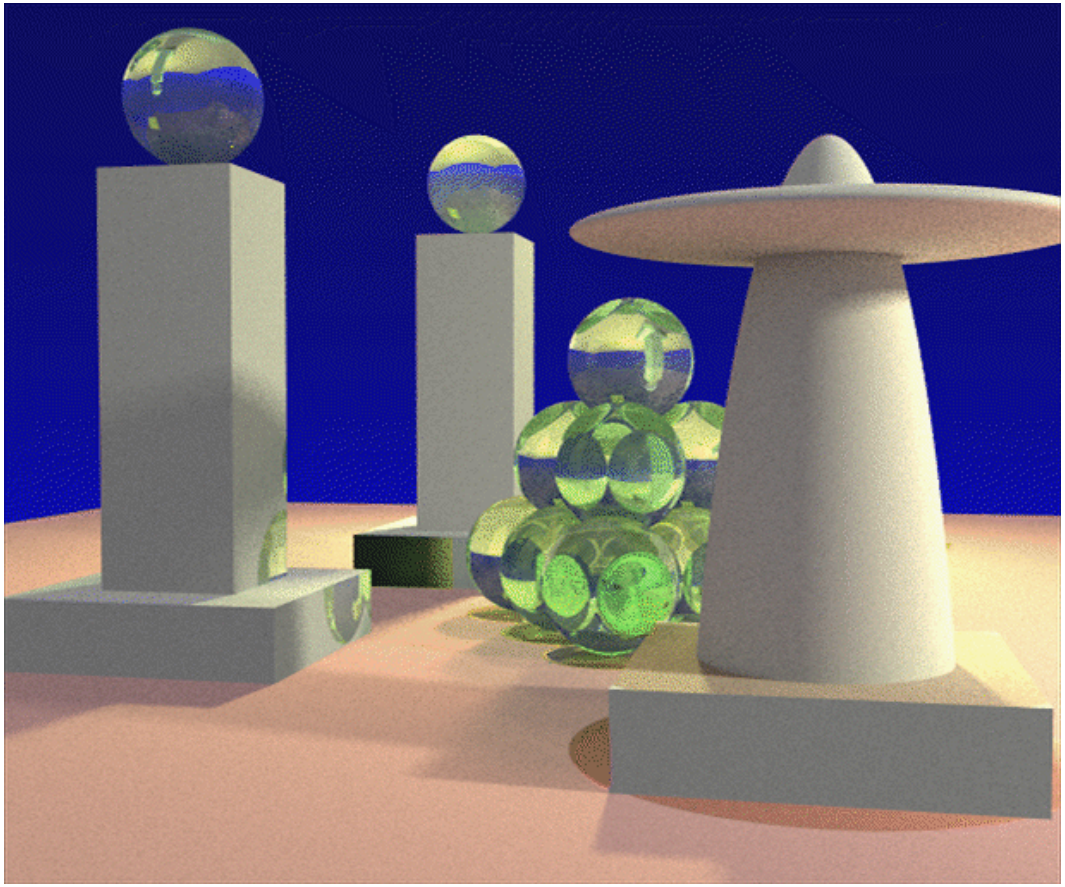
- ✓ lancé de rayon: [Whitted 80]
- ✓ radiosité (diffus): [Goral et al 84]
- ✓ Equation de rendu: [Kajiya 86]

Rendu: illumination globale (80's -)



- ✓ lancé de rayon: [Whitted 80]
- ✓ radiosité (diffus): [Goral et al 84]
- ✓ Equation de rendu: [Kajiya 86]

Rendu: illumination globale (80's -)



- ✓ lancé de rayon: [Whitted 80]
- ✓ radiosité (diffus): [Goral et al 84]
- ✓ Equation de rendu: [Kajiya 86]

Cette séance

Infos pratiques

Applications

Rapide historique

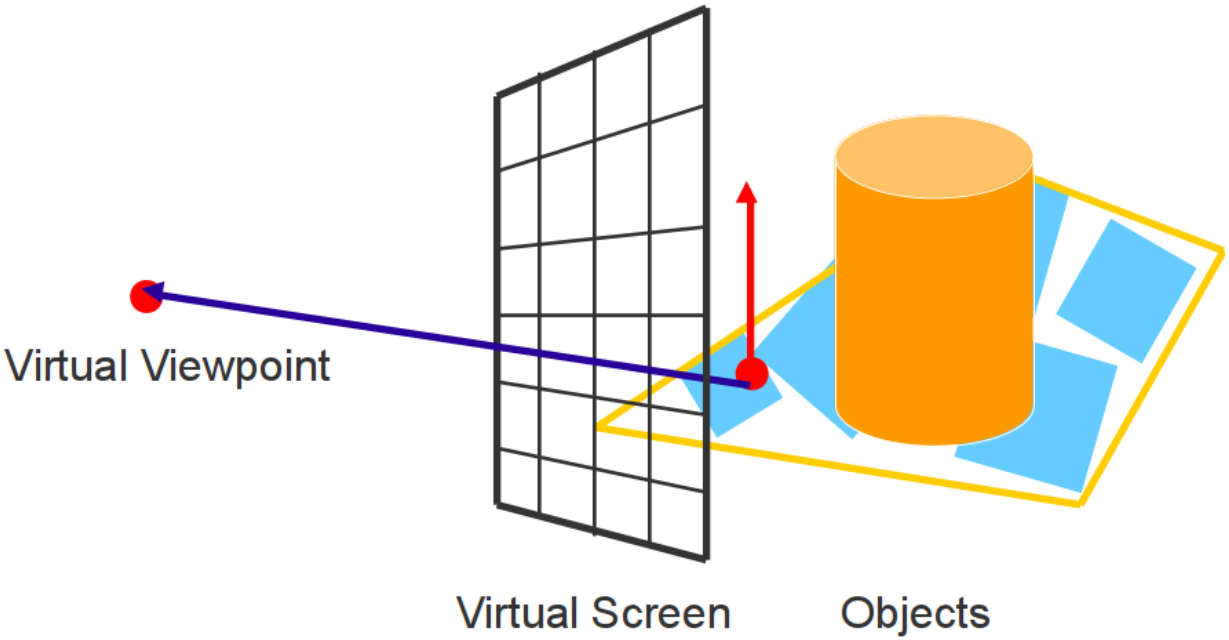
Pipeline
- Modélisation
- Animations
- Rendu

Thèmes abordés

Thèmes abordés (CM)

✓ **Rendu**

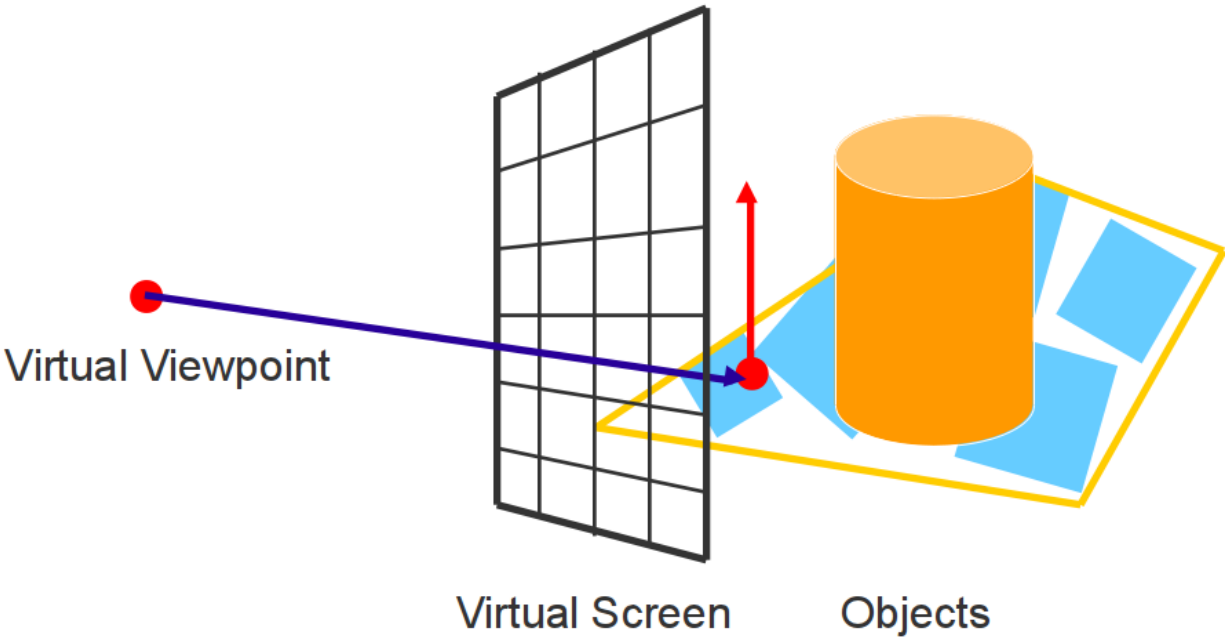
- modèles d'illumination directe à la OpenGL
- illumination directe par ray casting, lancé de rayon,
- couleurs: perception et modèles
- modèle physique: radiances, équation de rendu,
- radiosité, illumination globale



Thèmes abordés (CM)

✓ **Rendu**

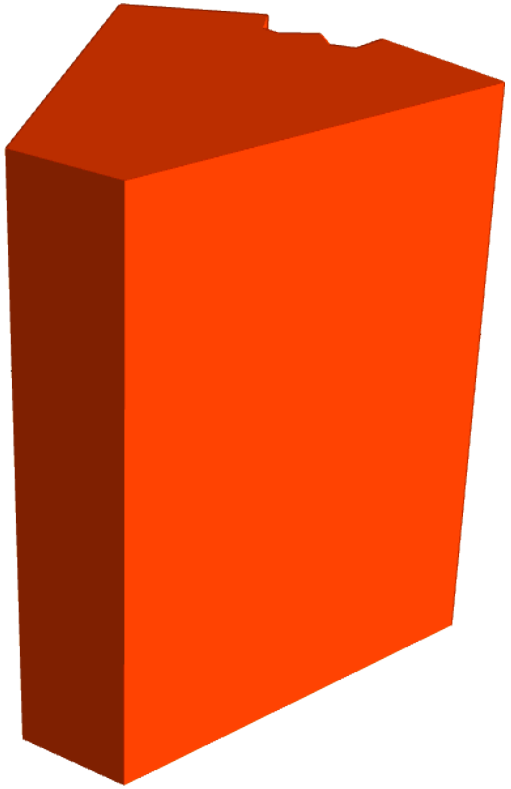
- modèles d'illumination directe à la OpenGL
- illumination directe par ray casting, lancé de rayon,
- couleurs: perception et modèles
- modèle physique: radiances, équation de rendu,
- radiosité, illumination globale



Thèmes abordés (CM)

✓ **Modélisation géométrique**

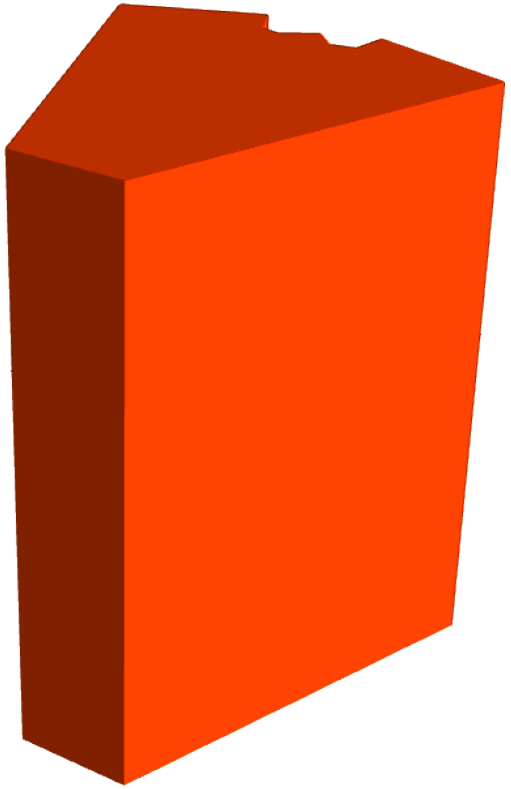
- modèles de surface (paramétrique, subdivision, nurbs, implicite,...)
- modèles procéduraux



Thèmes abordés (CM)

✓ **Modélisation géométrique**

- modèles de surface (paramétrique, subdivision, nurbs, implicite,...)
- modèles procéduraux



Thèmes abordés (CM)

✓ **Modélisation géométrique**

- modèles de surface (paramétrique, subdivision, nurbs, implicite,...)
- modèles procéduraux

Thèmes abordés (TP)

- ✓ OpenGL
  - Géométrie
  - Shading
  - Réflexions
  - Ombres
- ✓ Ray Tracer?

Thèmes abordés (TP)

- ✓ OpenGL
  - Géométrie
  - Shading
  - Réflexions
  - Ombres
- ✓ Ray Tracer?

```
typedef struct{double x,y,z}vec;vec U,black,amb={.02,.02,.02};struct sphere{ vec cen,color;
double rad,kd,ks,kt,kl,ir}*s,*best,sph[]={0.,6.,.5,1.,1.,1.,.9, .05,.2,.85,0.,1.7,-1.,8.,-.5,1.,.5,.2,1.,
.7,.3,0.,.05,1.2,1.,8.,-.5,.1,.8,.8, 1.,.3,.7,0.,0.,1.2,3.,-6.,15.,1.,.8,1.,7.,0.,0.,0.,.6,1.5,-3.,-3.,12.,
.8,1., 1.,5.,0.,0.,0.,.5,1.5,};yx;double u,b,tmin,sqrt(),tan();double vdot(A,B)vec A ,B;{return A.x
*B.x+A.y*B.y+A.z*B.z;}vec vcomb(a,A,B)double a;vec A,B;{B.x+=a* A.x;B.y+=a*A.y;B.z+=a*A.z;
return B;}vec vunit(A)vec A;{return vcomb(1./sqrt( vdot(A,A)),A,black);}struct sphere*intersect
(P,D)vec P,D;{best=0;tmin=1e30;s= sph+5;while(s-->sph)b=vdot(D,U=vcomb(-1.,P,s->cen)),
u=b*b-vdot(U,U)+s->rad*s ->rad,u=u>0?sqrt(u):1e31,u=b-u>1e-7?b-u:b+u,tmin=u>=1e-7&&
u<tmin?best=s,u: tmin;return best;}vec trace(level,P,D)vec P,D;{double d,eta,e;vec N,color;
struct sphere*s,*l;if(!level--)return black;if(s=intersect(P,D));else return amb;color=amb;eta=
s->ir;d= -vdot(D,N=vunit(vcomb(-1.,P=vcomb(tmin,D,P),s->cen )));if(d<0)N=vcomb(-1.,N,black),
eta=1/eta,d= -d;l=sph+5;while(l-->sph)if((e=l ->kl*vdot(N,U=vunit(vcomb(-1.,P,l->cen))))>0&&
intersect(P,U)==l)color=vcomb(e ,l->color,color);U=s->color;color.x*=U.x;color.y*=U.y;color.z
*=U.z;e=1-eta* eta*(1-d*d);return vcomb(s->kt,e>0?trace(level,P,vcomb(eta,D,vcomb(eta*d-
sqrt (e),N,black))):black,vcomb(s->ks,trace(level,P,vcomb(2*d,N,D)),vcomb(s->kd, color,vcomb
(s->kl,U,black))));}main(){printf("%d %d\n",32,32);while(yx<32*32) U.x=yx%32-32/2,U.z=32/2-
yx++/32,U.y=32/2/tan(25/114.5915590261),U=vcomb(255., trace(3,black,vunit(U)),black),printf
("%.0f %.0f %.0f\n",U);}/*minray!*/
```