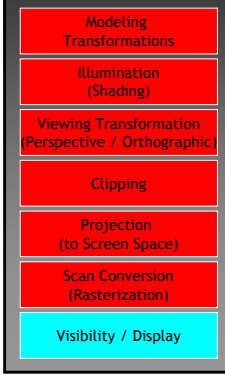
Plan

- Historique
- Pipeline graphique
- Transformations
- Image, couleur
- Rasterisation
- Visibilité
- Illumination

65

Visibilité, affichage



Calcul des primitives visibles : z-buffer.

Remplissage du frame buffer avec le bon format de couleur.

66

Qu'est-ce qu'une image ?

Image réelle : fonction à 2 variables une couleur.

Image numérique : tableau de pixels (picture element).

Informations nécessaires à la manipulation d'une image :
•nombre de lignes,
•nombre de colonnes,
•format des pixels (bits, niveaux de gris, de couleurs),
•compression éventuelle.

Très nombreux formats de fichiers images permettant de stocker ces informations ainsi que le tableau des valeurs : formats textes (PPM), binaires (BMP, GIF), binaires compressés avec (JPG) ou sans perte (PNG), …

67

Le « frame buffer »

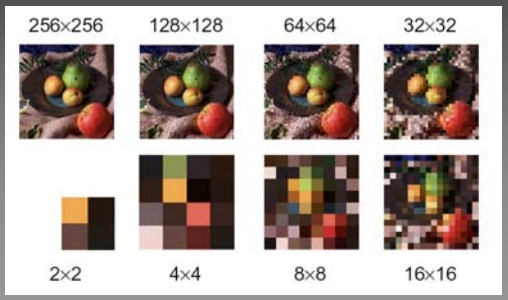
Partie de la mémoire réservée à l'affichage (Vidéo RAM).

Double buffer pour permettre un affichage plus fluide : on affiche un des buffer pendant que l'on met l'autre à jour.

Quadruple buffer pour faire de la stéréo en double buffer : deux buffer par œil.

68

Echantillonnage

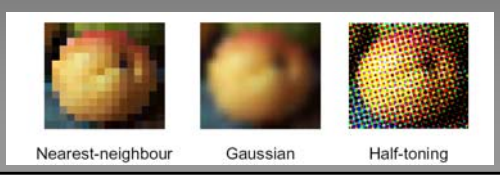


69

Affichage

La résolution du media de visualisation est donnée en dpi (dots per inch) : 100 pour un moniteur, 300-1200 pour une imprimante.

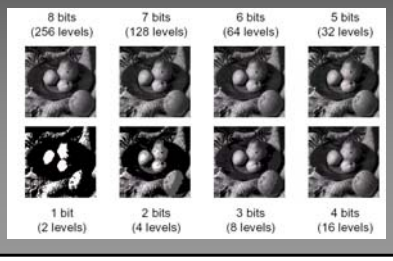
Suivant les cas on utilisera diverses méthodes d'affichage :



70

Intensité lumineuse

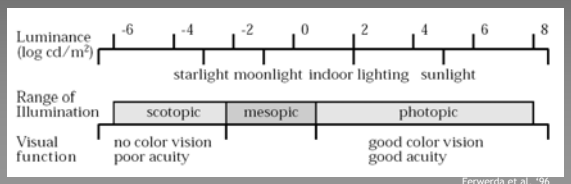
- L'intensité de chaque point doit aussi être échantillonnée : en général on utilise 8 bits pour coder le niveau d'intensité (niveau de gris) et donc 8*3 = 24 bit pour les couleurs (RVB).



71

Intensité lumineuse

- Avec 8 bits par couleur, 16m couleurs mais seulement 768 niveau de luminosité possibles
- Dans la réalité, rapport de luminosité de $1:10^{10}$ entre la nuit et le jour
- Sensibilité de l'œil logarithmique



Ferwerda et al. '96

72

Intensité lumineuse

- Formats d'image classiques : 24 bits

=> Besoin d'un codage HDR (High Dynamic Range)

- Sensibilité des appareils photos limitée

=> Besoin d'outils d'acquisition HDR

- Résolution des écrans limitée à 24 bits

=> Besoin d'outils d'affichage HDR

73

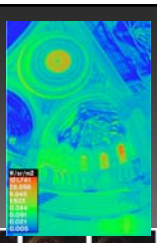
Intensité lumineuse

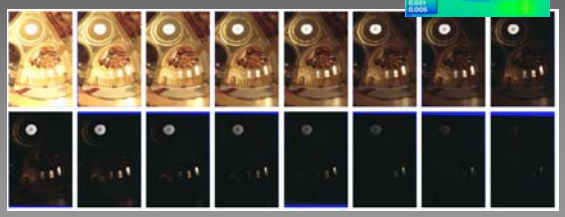
- Formats d'image HDR
  - SGI LogLuv 32 bits TIFF
  - OpenEXR
  - ...

74

Intensité lumineuse

- Acquisition HDR
  - A partir d'appareils photos standards
  - A partir de caméras HDR (récent)

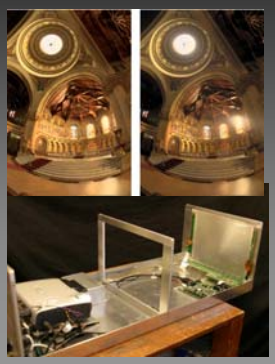




75

Intensité lumineuse

- Rendu HDR
  - Sur un écran standard, en simulant la sensibilité de l'œil (Tone Mapping)
    - Contrast & brightness
    - Color balance
    - Low vision
    - Glare
    - Motion blur
    - Lens flare
  - Sur écran HDR (expérimental)



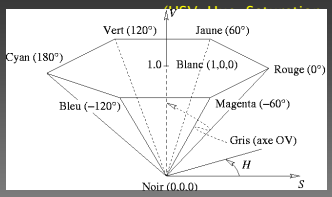
76

Qu'est-ce-que la couleur ?

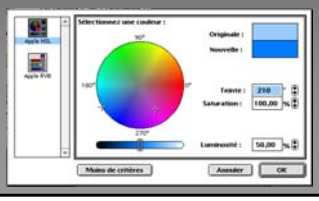
- Qu'est-ce qu'une couleur ? Définitions
  - Artistiques
    - Teinte, saturation, luminance
  - Physiques/biologiques
    - Spectre, stimulus
    - Fonctions de base universelles
    - Espaces perceptuellement uniformes
  - Informatiques
    - RGB, CMYK, HSV…

77

Modèle du peintre : TLS = Teinte, Luminance, Saturation



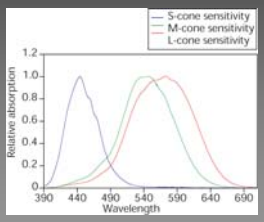
Pratique pour les interfaces graphiques



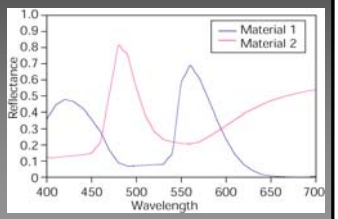
78

Représentation spectrale : spectre = fonction de la longueur d'onde dans le domaine visible.

Des spectres différents donnent la même couleur car l'oeil n'a que 3 types de capteurs : les cones.



Cones

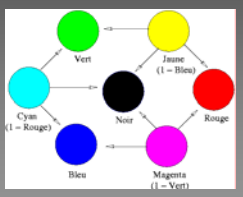


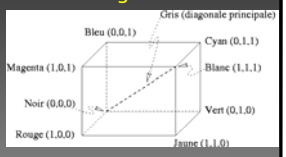
2 couleurs identiques pour l'oeil

79

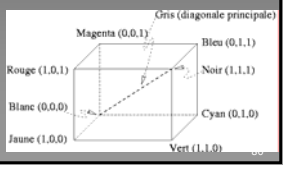
Echantillonnage spectral : risque de perte d'information si on ne choisi pas un bon jeu d'échantillon. Très bons résultats avec 4 échantillons choisis ou 9 échantillons réguliers.

RVB : additif, système visuel humain, écrans.



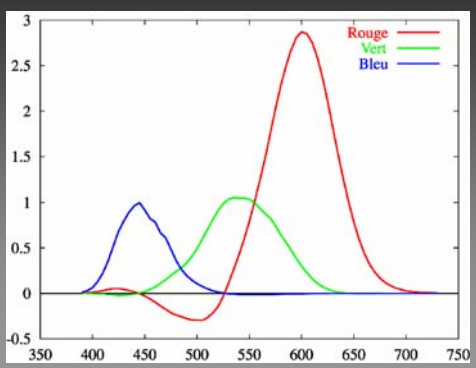


CMJ : soustractif, peinture, imprimantes.



80

RVB : Un léger défaut



81

Une nouvelle fonction de base

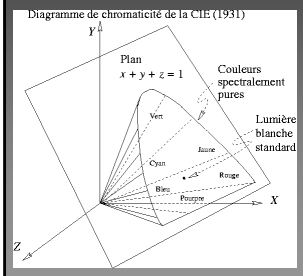
- Les couleurs primaires ont un défaut :
  - L'ensemble des couleurs visibles ne peut pas être représenté avec des coordonnées positives
- Besoin de nouvelles fonctions de base
  - Couvrant tout le visible
  - Avec des coordonnées positives
  - Linéaires par rapport à RVB
- Commission Internationale de l'Éclairage
  - www.cie.co.at
  - 1931

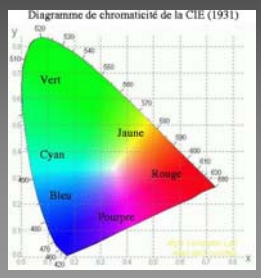
82

Modèle de la Commission Internationale de l'Éclairage (CIE) : 3 primaires standard X, Y, Z sans coeff négatifs.

La couleur est donnée par x, y, z :

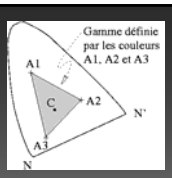
$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z} \quad \therefore \quad x+y+z = 1$$





83

La gamme de couleur d'une imprimante est généralement plus petite que celle du moniteur
=> les couleurs vues à l'écran ne peuvent pas toutes être rendues en impression
=> il faut réduire la gamme du moniteur



Il n'existe pas de base trichromique de couleurs visibles qui permette de couvrir l'ensemble des couleurs visibles.

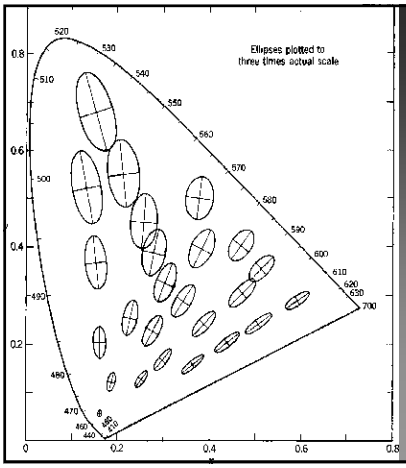
Il est donc nécessaire d'avoir recours à des composantes soustractives pour couvrir le spectre visible dans les modèles tels que RVB.

84

Perception des couleurs

- Distance entre deux couleurs :
  - Dans l'espace de base : facile
  - Pour la vision humaine : utile
- Idéalement, il y a un lien entre les deux
- Espace des couleurs *perceptuellement uniforme*
  - Lien constant, indépendant de la couleur
- Différences juste perceptibles :
  - Plus petite distance entre deux couleurs perçues comme différentes

85



Différences juste perceptibles dans l'espace *xy*

86

Espaces perceptuellement uniformes

- CIE, 1976
- L*a*b* et L*u*v*

$$L^* = 116\left(Y/Y_n\right)^{\frac{1}{3}} - 16 \text{ si } Y/Y_n > 0.008856$$
$$L^* = 903.3\left(Y/Y_n\right) \text{ sinon}$$
$$a^* = 500\left[\left(X/X_n\right)^{\frac{1}{3}} - \left(Y/Y_n\right)^{\frac{1}{3}}\right]$$
$$b^* = 200\left[\left(Y/Y_n\right)^{\frac{1}{3}} - \left(Z/Z_n\right)^{\frac{1}{3}}\right]$$

$$u^* = 13L^*(u' - u'_n)$$
$$v^* = 13L^*(v' - v'_n)$$
$$u' = \frac{4X}{X+15Y+3Z}$$
$$v' = \frac{9Y}{X+15Y+3Z}$$

A_n : coordonnée d'un blanc de référence

- En pratique, ce n'est pas parfaitement conforme à la perception => il y a encore de la recherche…

87

Résumé : Espaces de couleur

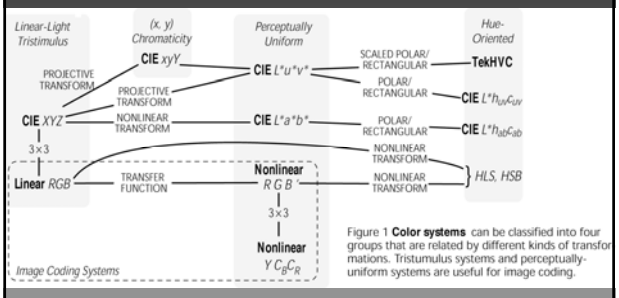


Figure 1 **Color systems** can be classified into four groups that are related by different kinds of transformations. Tristimulus systems and perceptually-uniform systems are useful for image coding.

88

Plan

- Historique
- Pipeline graphique
- Transformations
- Image, couleur
- Rasterisation
- Visibilité
- Illumination

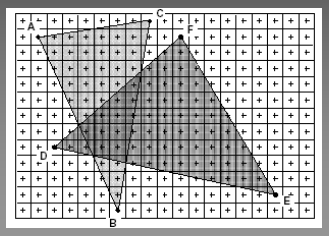
89

Rasterisation

Modeling Transformations
Illumination (Shading)
Viewing Transformation (Perspective / Orthographic)
Clipping
Projection (to Screen Space)
Scan Conversion (Rasterization)
Visibility / Display

Découpe la primitive 2D en pixels

Interpole les valeurs connues aux sommets : couleur, profondeur,…



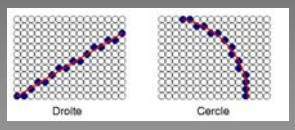
90

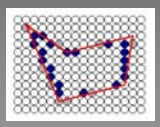
Primitives 2D

Dans tous les cas il faut savoir afficher du 2D, ligne ou polygone

⇒ il faut savoir quels pixels allumer pour un objet mathématique.

⇒ il faut savoir remplir un polygone.

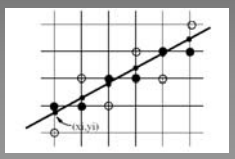




91

Tracé de droites

- Méthodes brutes (naïve) : discrétisation de la primitive en n points, puis approximation au pixel le plus proche pour chacun des n points. Peu efficace et peu précise.
- Méthodes incrémentales : La position du point suivant est choisie de façon à minimiser l'erreur d'approximation. Méthode optimale pour le tracé de segments de droites.



92

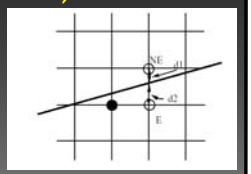
Algorithmes de Bresenham (1965)

Premier octant :
Une droite est définie par l'équation:
y = mx +B

On cherche le prochain pixel comme celui qui minimise l'erreur
e = (d2 - d1)/2

Au premier point (x0+1, y0+m),
e1 = (d2 - d1)/2 = - 0.5;

Ensuite l'erreur se propage :
NE : e = e + m - 1;
E : e = e + m.



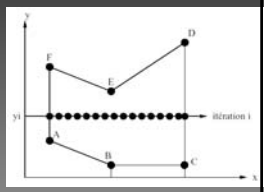

```
void Bresenham(x0, y0, x1, y1, valeur)
{ int x;
  double m = y1 - y0 / x1 - x0 ;
  double e = -0.5;
  int y = y0;
  for(x = x0; x <= x1 ; x++)
  { e = e  + m ;
    AfficherPixel(x, y, valeur);
    if (e >= 0)
    { y = y + 1;
      e = e - 1;
}}}
```

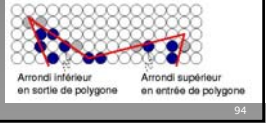
93

Remplissage de polygone

Principe (scan-line)

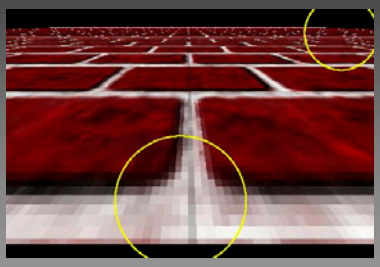
1. Remplissage par recherche des points d'intersection d'une ligne horizontale avec des contours (un nombre pair).
2. Une fois les points d'intersection obtenus, remplissage selon une règle de parité : incrémentation de la parité à chaque traversée de frontière et tracé si impair.
3. Gestion des conflits frontaliers : on prend les points intérieurs au polygone pour éviter les chevauchements.





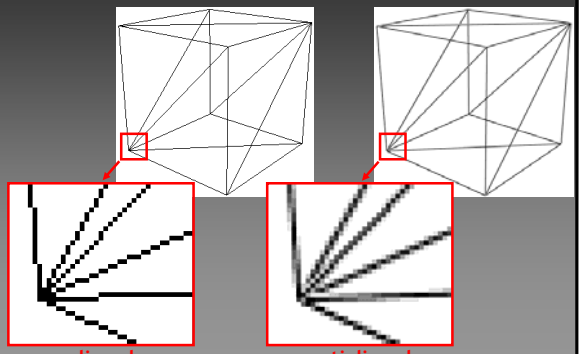
94

Texturing – Filtering



95

Aliasing



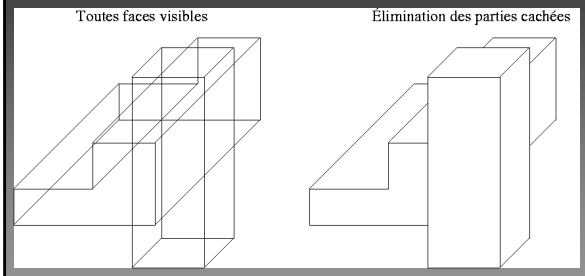
96

Plan

- Historique
- Pipeline graphique
- Transformations
- Image, couleur
- Rasterisation
- Visibilité
- Illumination

97

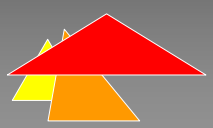
Élimination des parties cachées



98

Une scène = un ensemble de primitives :
En général un ensemble de polygones (triangles)

Détermination des surfaces visibles :
équivalent au tri => O(*n* log *n*)



99

Algorithme à précision objet : le masquage se fait sur le modèle de données physiques. => Calcul en coordonnées du monde.

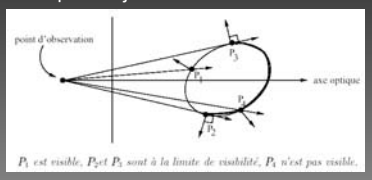
Avantage : le masquage des parties cachées est valable quelle que soit l'échelle du dessin.
Inconvénient : plus cher.

Algorithme à précision image : le masquage se fait sur les pixels écran. => Calcul en coordonnées fenêtre.

Avantage : peut utiliser la cohérence, plus rapide en moyenne
Inconvénient : précision limitée au nombre de pixels

100

Backface culling : éliminer les parties de la surface pour lesquelles la normale pointe dans la direction opposée au point d'observation. Ces parties, vues du point d'observation, sont en effet occultées par l'objet lui-même.



P_1 est visible, P_2et P_3 sont à la limite de visibilité, P_4 n'est pas visible.

La suppression des faces arrières suffit à éliminer les parties cachées :

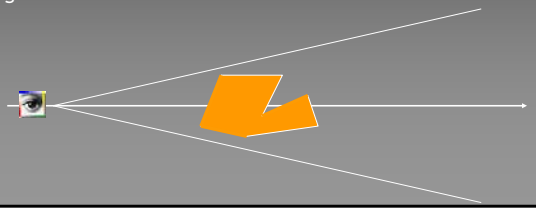
1. si l'objet est **seul dans la scène** (il ne faut pas qu'un objet puisse en masquer un autre);
2. si l'objet est **convexe** (dans un objet concave, des faces avant peuvent être masquées par d'autres).

101

Calcul : produit scalaire

- (Sommet-PointDeVue)*normale
- > 0 : on garde le polygone
- < 0 : on l'élimine

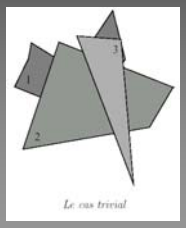
Économise 50 % du temps de calcul => préalable aux autres algos

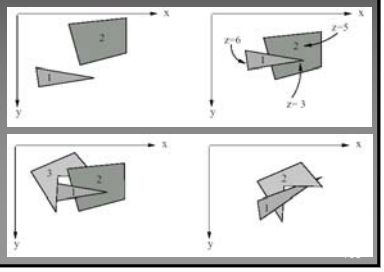


102

Algorithme du peintre : peindre les facettes polygonales dans la mémoire vidéo suivant un ordre de distance décroissante au point d'observation.

1. Trier les facettes suivant les z décroissants dans le repère de la camera.
2. Résoudre les ambiguïtés dans la liste lorsque les facettes se recouvrent.
3. Projeter les facettes et remplir les polygones suivant la liste.





Pour ou contre ?

- Le plus intuitif des algorithmes
- Coût en mémoire :
  - Affichage direct à l'écran : O(*p*)
  - Il faut trier les polygones : O(*n*log*n*)
- Temps de calcul :
  - On affiche toute la scène
  - Efficace surtout sur des petites scènes

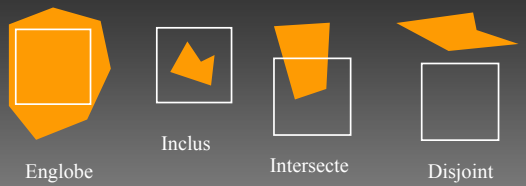
104

Area Subdivision (Warnock)

- On utilise la cohérence spatiale
- On divise l'écran en zones de travail
- Pour chaque zone, on ne considère que les polygones qui l'intersectent
- Si la visibilité est connue, on s'arrête
- Si la visibilité est inconnue, on subdivise
- On interrompt la subdivision quand on atteint une taille limite

105

Polygones/zone de travail

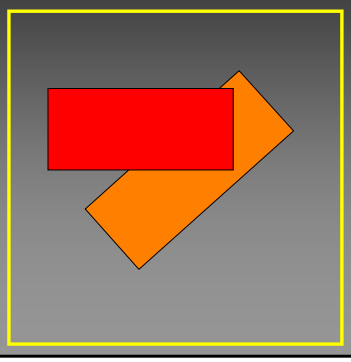


Quand est-ce que la visibilité est connue ?

- Tous les polygones sont *disjoints*
- Un seul polygone *intersecte* ou *inclus*
- Un polygone *englobant* qui est devant les autres

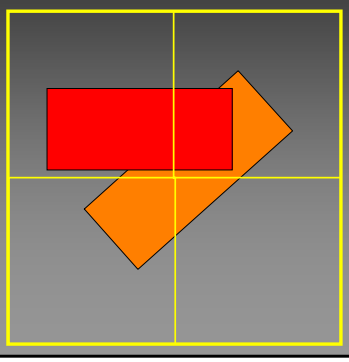
106

Warnock : exemple (1)



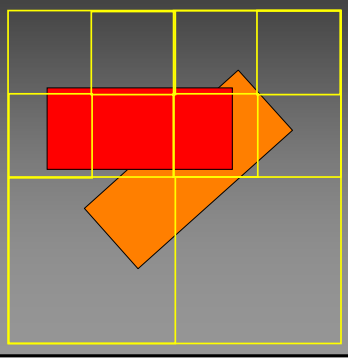
107

Warnock : exemple (2)



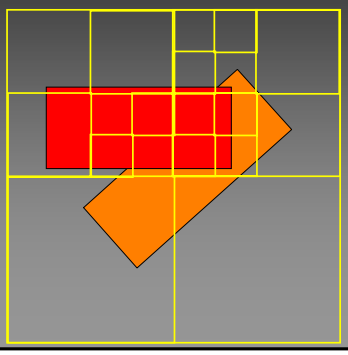
108

Warnock : exemple (3)



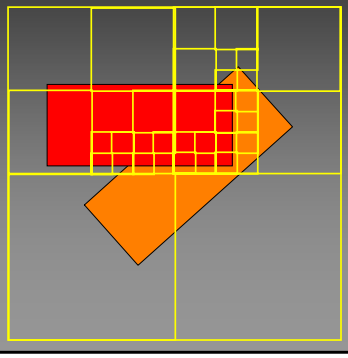
109

Warnock : exemple (4)



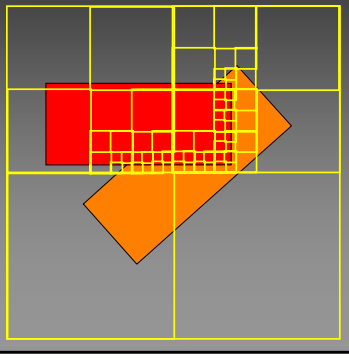
110

Warnock : exemple (5)



111

Warnock : exemple (6)



112

Jean-Dominique.Gascuel@inrialpes.fr

Warnock : pour ou contre

- Utilise la cohérence spatiale
- Plus efficace avec des grands polygones
- Coût mémoire parfois élevé
- Implémentation facile : appels récursifs à la même fonction

113

Z-Buffer

- Un tableau, de la taille de l'écran
- On stocke la valeur maximale de z pour chaque pixel
  - *z* est la direction de visée, exprime la distance à l'oeil
- Initialisation : tous les pixels à moins l'infini
- Projection de tous les polygones
  - On met à jour les pixels de la projection du polygone

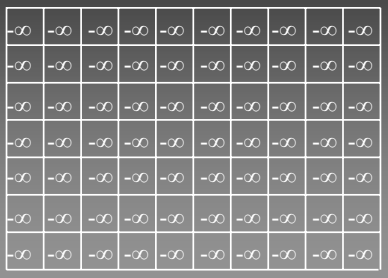
114

Z-buffer : algorithme

- Pour chaque polygone :
  - Projeter le polygone sur le plan image
  - Pour chaque pixel dans la projection du polygone
    - Calculer la valeur de *z* pour ce pixel
    - Si z est supérieur à la valeur courant de z max
      - Changer z maximal
      - Afficher le pixel à l'écran, de la couleur du polygone

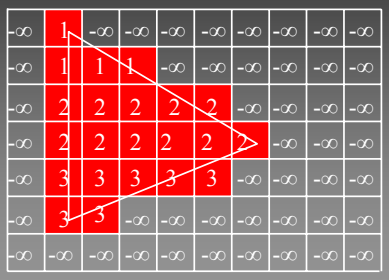
115

Z-buffer (1)



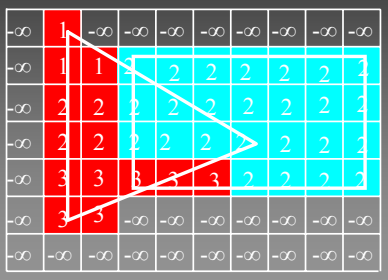
116

Z-buffer (2)



117

Z-buffer (3)



118

Z-Buffer : pour ou contre

- Pour :
  - Facile à implémenter, scan-line
  - Travaille dans l'espace image
    - Rapide
- Contre :
  - Coût en mémoire
  - Travaille dans l'espace image
    - aliasing
    - artefacts

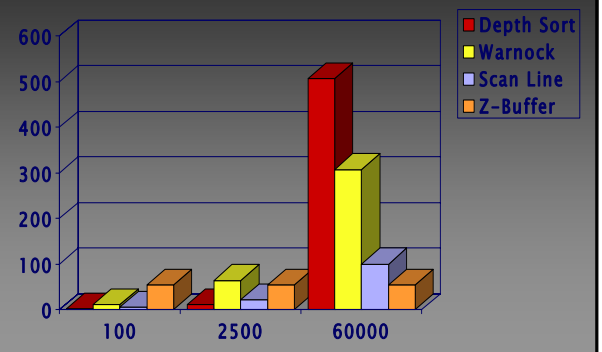
119

Z-Buffer

- Combien d'information ?
  - Combien de bits pour z max ?
  - Limité par la mémoire :
    - 8 bits, 1024x1280: 1.25 Mb
    - 16 bits, 1024x1280: 2.5 Mb
  - Nécessaire pour la séparation des objets proches :
    - 8 bits, distance minimale entre objets de 0.4 % (4mm pour 1m)
    - 16 bits, distance minimale de 0.001 % (1mm pour 1km)
    - Que se passe t-il en dessous de cette limite ?

120

Coûts comparés



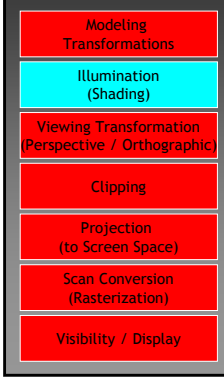
121

Plan

- Historique
- Pipeline graphique
- Transformations
- Image, couleur
- Rasterisation
- Visibilité
- Illumination

122

Illumination



Les primitives sont éclairées selon leur matériau, le type de surface et les sources de lumière.

123

Illumination

Quelle couleur doit-on afficher pour ce pixel ?

=> Résultat de l'interaction de la lumière avec la scène et l'oeil.

Dépend donc de :

- position du point dans l'espace,
- orientation du point (élément de surface),
- caractéristiques de la surface (diffusion, réflexion, transparence…),
- sources de lumière (surfaciques, ponctuelles…).
- Position et orientation de la "caméra"

124

La lumière : onde et corpuscule => propagation et transport.

Un photon transporte une certaine énergie à une longueur d'onde donnée. A chaque interaction (réflexion, transmission, réfraction, absorption, diffraction et interference) il peut changer sa direction et sa couleur (spatial et spectral).

Si un photon passe par la position de l'oeil tout en intersectant la fenêtre graphique, alors sa couleur contribue au pixel qu'il traverse.

⇒L'image parfaite demande de suivre une infinité de photons, il faut donc simplifier le problème :

Illumination locale, ombrage, illumination globale.

125

Modèles d'illumination locale

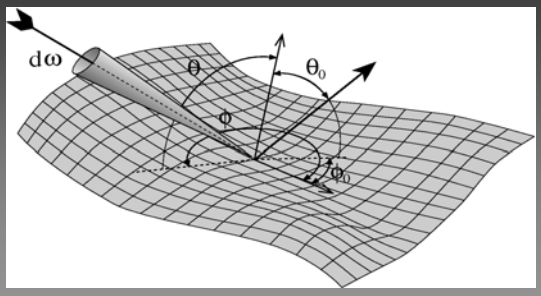
On suppose que les objets interagissent avec une seule source de lumière ponctuelle.

On ne calcule pas les interactions entre objets => pas d'ombres, pas d'effet miroir…

On calcule la couleur en chaque point.

126

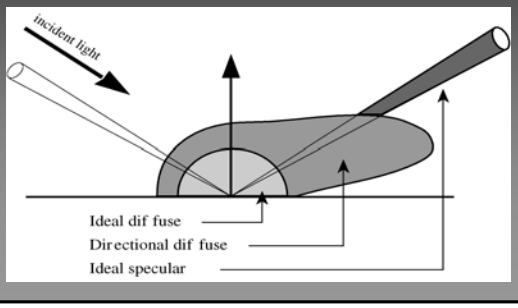
BRDF : *Bi-directional Reflectance Distribution Function*



127

Rapport entre la quantité de lumière reçue et émise;

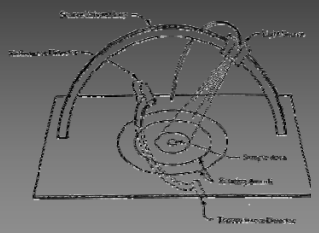
Décrit complètement le comportement de la surface en chaque point et pour chaque direction d'émission et de réception.



128

Modèle complexe et difficile à obtenir

Il existe des systèmes de saisie :



129

Modèles simplifiés

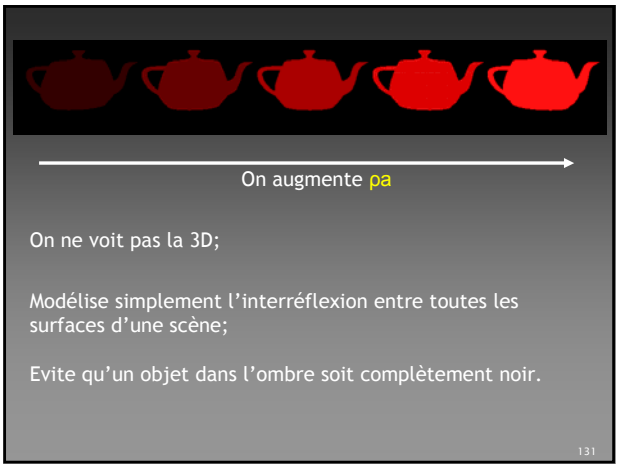
Lumière ambiante : une source lumineuse non ponctuelle qui émet de manière constante dans toutes les directions et sur toutes les surfaces une lumière d'intensité Ia.

Alors pour un point P de la surface, l'intensité lumineuse sera

I(P) = ρa Ia;

qui est constante en tous points de la surface, et où ρa est un facteur qui détermine la quantité de lumière ambiante réfléchie par la surface et est fonction des propriétés matérielles de la surface (0 <= ρa <= 1).

130



On ne voit pas la 3D;

Modélise simplement l'interréflexion entre toutes les surfaces d'une scène;

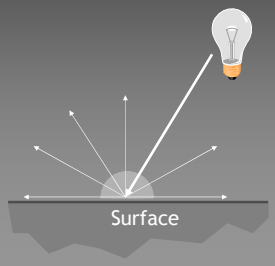
Evite qu'un objet dans l'ombre soit complètement noir.

131

Réflexion diffuse : source lumineuse ponctuelle qui émet de manière constante dans toutes les directions.

Les surfaces Lambertiennes : surfaces mates (craie, papier), l'intensité en un point de la surface dépend uniquement de l'angle entre la normale à la surface et la direction du point à la source lumineuse.

BRDF uniforme



132

$$I(P) = \rho d \quad I \quad \cos \theta$$

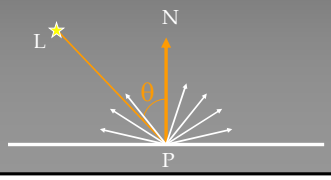
$$I(P) = \rho d \quad I \quad NP \bullet LP$$

I : intensité de la source lumineuse ponctuelle;

ρd : coefficient de réflexion diffuse de la surface (propriété matérielle : 0 <= ρd <= 1);

NP : la normale à la surface au point P;

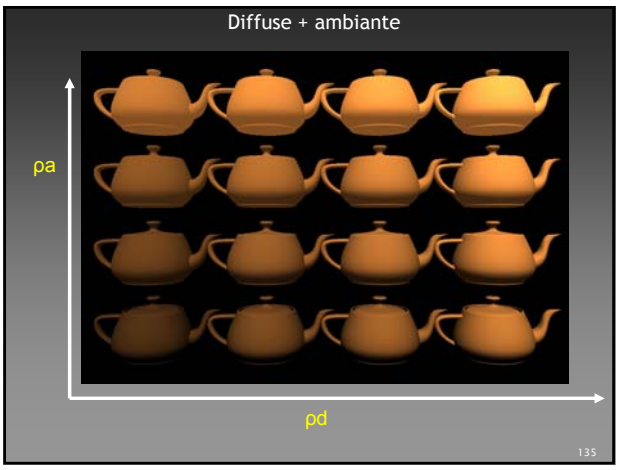
LP : la direction du point P à la source lumineuse.



133

134

Diffuse + ambiante



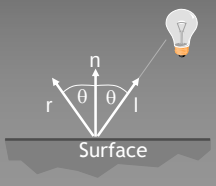
135

Réflexion spéculaire :

Surfaces brillantes (miroir).

Loi de Snell / Descartes : la lumière qui atteind l'objet est réflechie dans la direction ayant le même angle.

BRDF : distribution de Dirac



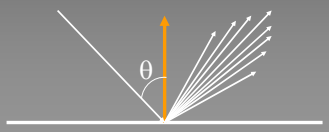
136

Problème : avec une source de lumière ponctuelle, l'effet n'est visible que dans une direction.

C'est utile pour l'illumination indirecte (ombres, miroirs) mais inutilisable pour calculer la couleur des pixels.

⇒ On suppose que la surface n'est pas parfaitement spéculaire.

⇒ Modèle de Phong.

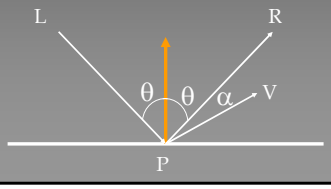


137

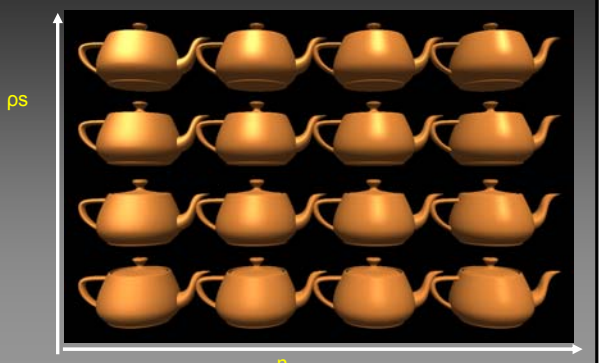
Modèle de Phong [1975] : la réflexion est importante lorsque l'angle α entre la direction d'observation du point de la surface V et la direction de réflexion R (symétrique de la direction LP par rapport à NP) est faible. Cette réflexion diminue de façon importante lorsque l'angle augmente.

$$I(P) = \rho s \quad I \quad \cos^n \alpha$$

n = rugosité : ∞ (1024) pour un miroir, 1 pour une surface très rugueuse.



138



139

Modèle complet : on ajoute tout, on pondère avec un coefficient d'atténuation Fd :

$$I(P) = \rho a \quad Ia \quad + \quad Fd \quad I \quad (\quad \rho d \quad NP \bullet LP \quad + \quad \rho s \quad \cos^n \alpha \ )$$

Modèle coloré : une intensité par composante de couleur.

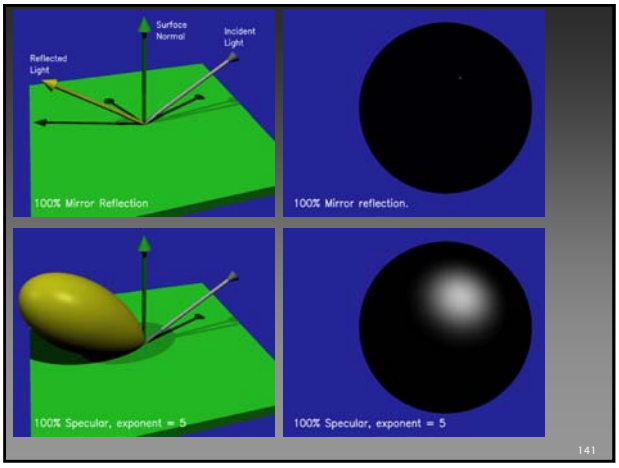
Plusieurs sources lumineuses : somme des intensités.

Transparence : manière de combiner couleur de fond et couleur de l'objet. Un paramètre de transparence t.

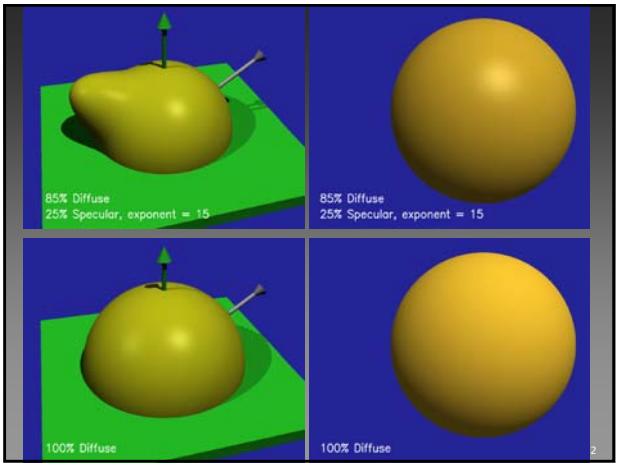
$$I = t.I(P) + (1-t).I(\text{derriere P})$$

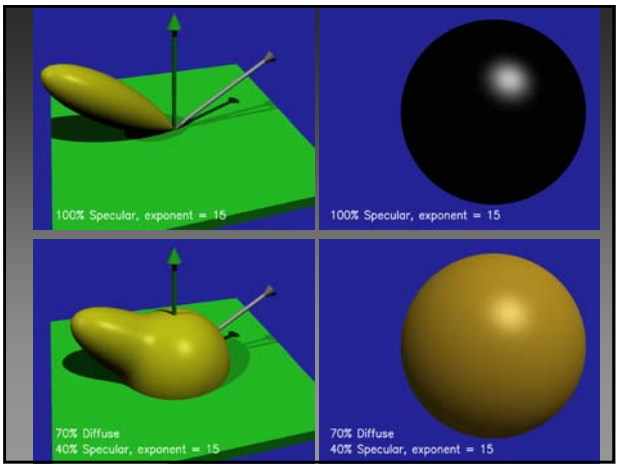
Halo : la couleur dépend de l'épaisseur traversée.

140



141





Cook-Torrance-Sparrow Model

Une surface est constituée de microfacettes.

La lumière arrivant sur une facette subit toutes les interactions possibles.

On fait une étude statistique qui dépend de la répartition des micro-facettes pour obtenir une BRDF valable sur toute la surface.

Il n'y a pas de formule simple mais ce modèle approxime très bien les effets physiques des matériaux.

144

Modèles d'interpolation (shading)

Flat shading : pour les facettes polygonales. Calculer une seule valeur d'illumination pour l'ensemble de la facette. Par exemple au point milieu de la facette en prenant pour normale à la surface celle du plan contenant la facette.

Cette approche est valide par rapport aux modèles d'illumination vus précédemment lorsque :

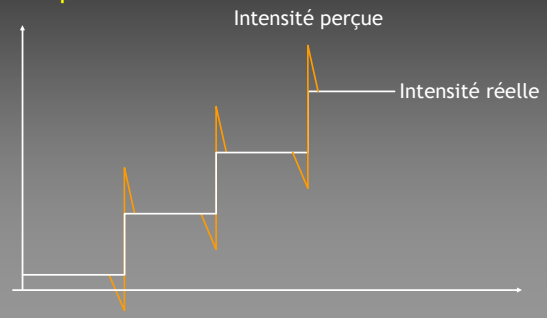
- la source lumineuse est à l'infini,
- la projection est orthographique,
- la surface est composée de facettes polygonales uniquement.

145

Problème : il y a des discontinuités le long des facettes;

L'oeil les voit très bien : effet Mach Banding;

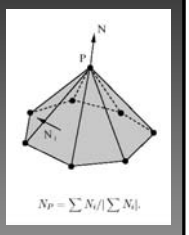
⇒ **Interpolation.**

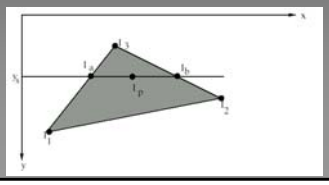


146

Gouraud shading : élimine les discontinuités d'intensité sur une facette polygonale par interpolation des valeurs d'intensité aux sommets de la facette.

1. Calcul des normales aux sommets;
2. Calcul des intensités aux sommets des facettes polygonales;
3. Interpolation par un algo de balayage.

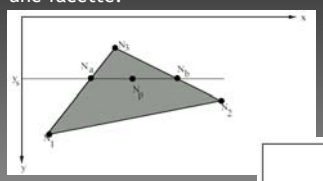


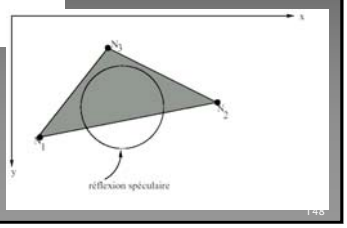


$$I_a = I_3 - (I_3 - I_1)(y_3 - y_s)/(y_3 - y_1),$$
$$I_b = I_3 - (I_3 - I_2)(y_3 - y_s)/(y_3 - y_2),$$
$$I_p = I_b - (I_b - I_a)(x_b - x_p)/(x_b - x_a).$$

Phong shading : calcul des normales par interpolation

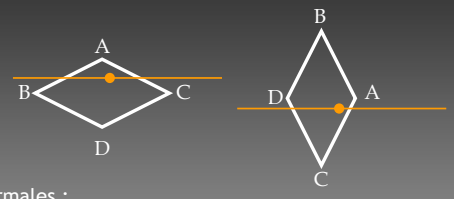
=> Permet de traiter les effets spéculaires contenus dans une facette.



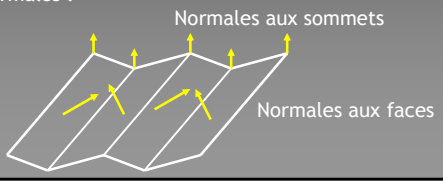


Problèmes

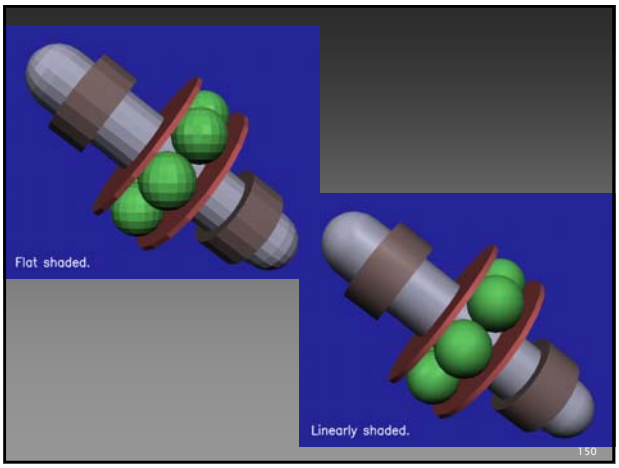
- Interpolation par balayage : perspective, rotation;



- Normales :



149



150

Résumé

- Modeling Transformations
- Illumination (Shading)
- Viewing Transformation (Perspective / Orthographic)
- Clipping
- Projection (to Screen Space)
- Scan Conversion (Rasterization)
- Visibility / Display

Rendu : Il nous reste à voir les traitements que l'on peut faire sur la scène pour accélérer le rendu ou pour améliorer le résultat visuel

Textures et matériaux

Rendu temps-réel

- -Calcul des ombres
- -Visibilité
- -Niveaux de détails
- -Image-based rendering

Illumination globale

Rendu expressif

151