



Rendu dans l'espace image et Textures

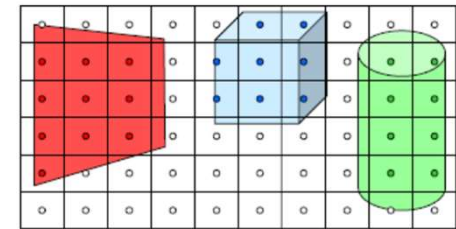
Céline Loscos

Introduction

- Nous allons répondre à la question :
 - Quel est le procédé qui permet l'affichage de la scène 3D sur l'image selon le point de vue de la camera ?
- Autrement dit, nous devons déterminer pour chaque pixel la couleur qui devra être affichée selon la façon dont un polygone est vue depuis le point de vue de la caméra

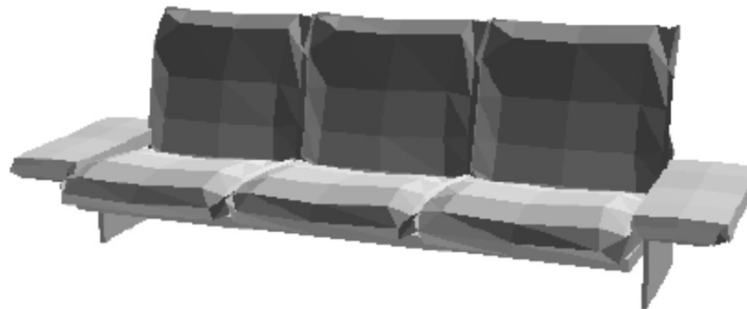
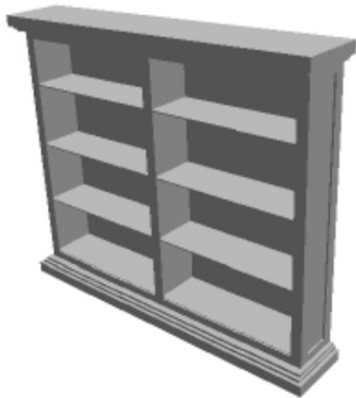
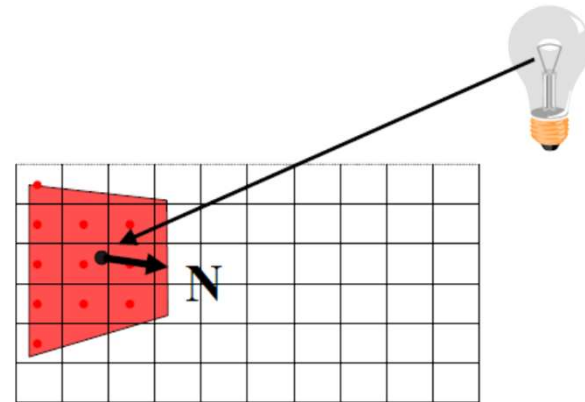
Introduction

- Une façon commune de le faire est par lancer de rayons
 - Algorithme
 - Tracer un rayon depuis le centre de la camera au travers de chaque pixel
 - Intersecter ce rayon avec les polygones de la scène
 - Trouver le polygone visible et calculer la couleur à afficher
 - Procédé coûteux, surtout s'il est réalisé de façon sophistiquée pour obtenir une image photoréaliste
- Autre considération : projeter les polygones sur l'image et prendre les décisions de la couleur des pixels dans l'espace image



Flat shading

- Calcul pour le polygone,
 - Affichage de la même couleur pour tous les pixels couverts par le polygone
- Problème : pas de lissage



Contenu

- Gestion de la profondeur
- Gouraud shading
- Phong shading
- Texture

Gestion de la profondeur

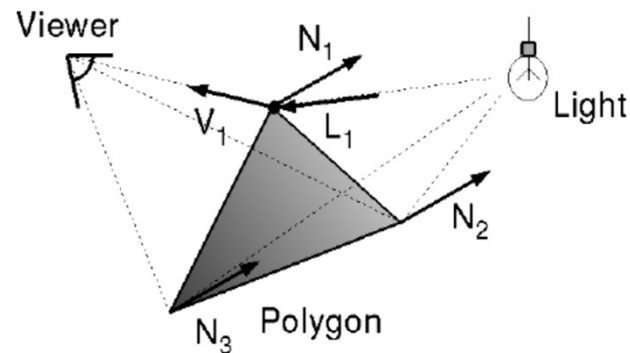
- Lorsque les polygones sont projetés sur la fenêtre de vue (l'image), ils peuvent se superposer
- Il faut trouver quel polygone est devant les autres
- Plusieurs méthodes existent pour gérer ce problème
 - Buffer de profondeur selon une ligne de scan
 - Z-buffer
 - Subdivision récursive
- Notation :
 - Z : profondeur (depth en anglais)
- (Non développé dans ce cours)

Gouraud Shading – étape 1

- Reprenons l'équation du modèle d'éclairage pour une réflexion diffuse I_{rp} en un point p

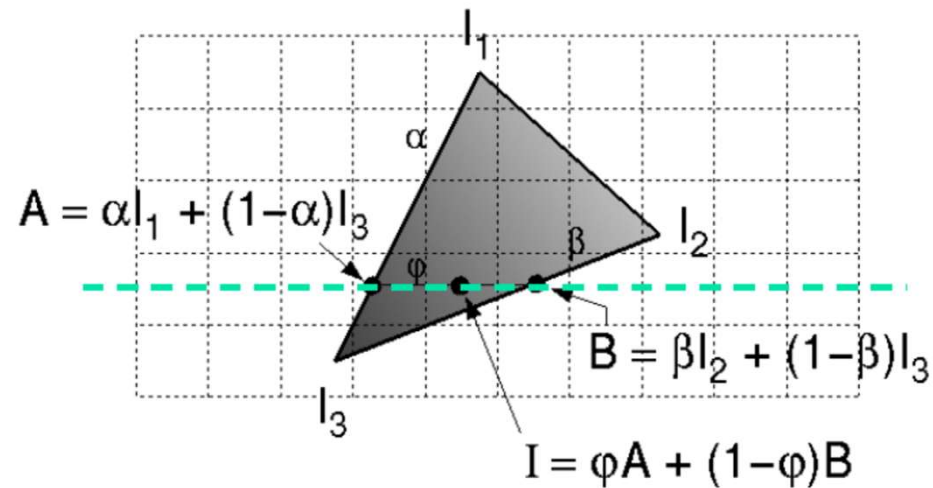
$$I_{rp} = k_a I_a + k_d \sum_{j=1}^q I_{pj} \cdot (n \cdot l_j)$$

- Calcul pour chaque sommet p du polygone



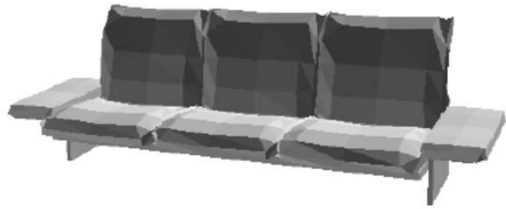
Gouraud shading – étape 2

- Gouraud interpolate la couleur en suivant les cotés et le long des lignes de scan de la fenêtre (interpolation bilinéaire)

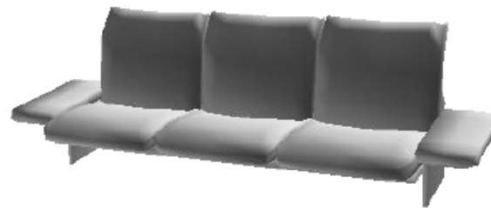


Gouraud shading - résultat

- Résultats : lissage des valeurs d'éclairage



Flat Shading



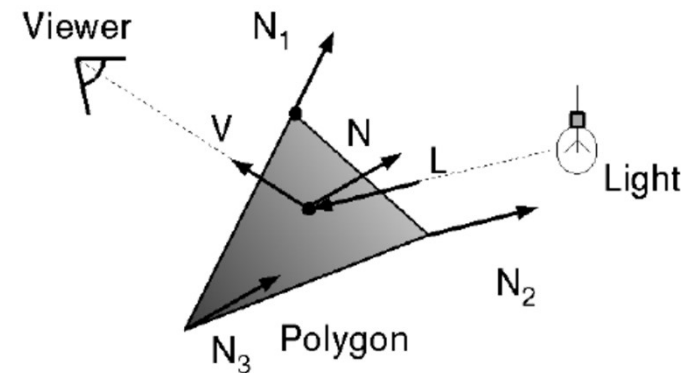
Gouraud Shading

- Problèmes
 - Couleur non constante lors de rotation
 - Manque les taches spéculaires

Phong Shading

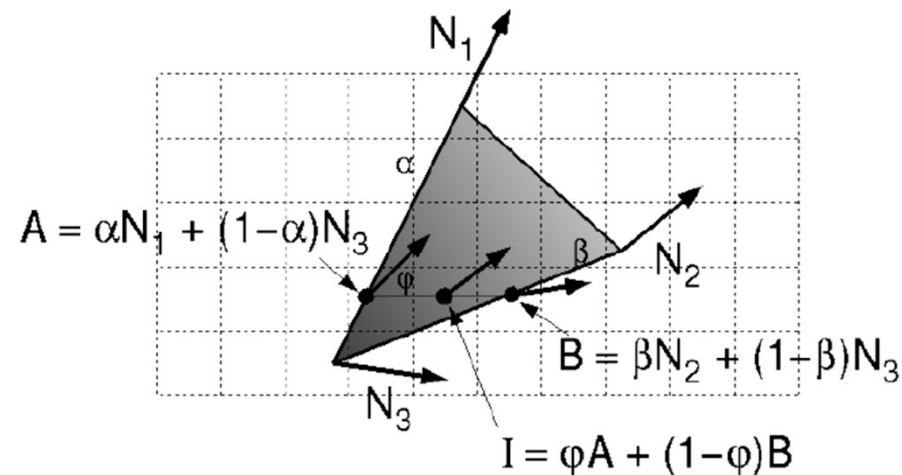
$$I_p = k_a I_a + \sum_{i=1}^N I_{pi} \cdot ((n \cdot l_i)k_d + (h_i \cdot n)^m k_s)$$

- Inclus le composant spéculaire
- Comment ?
 - En interpolant les normales au lieu des couleurs !
- Recrée les tâches spéculaires au centre des polygones



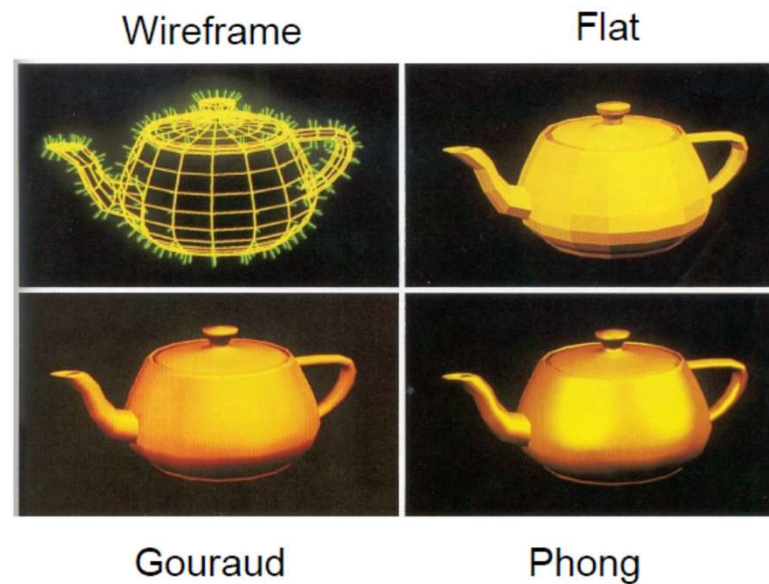
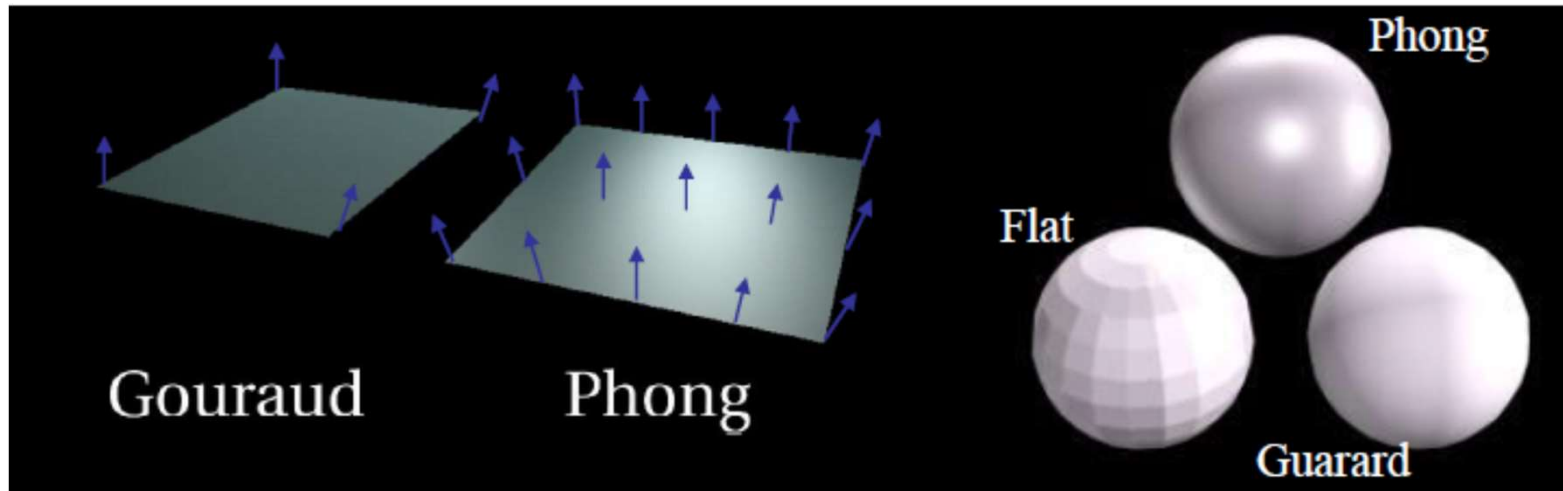
Phong shading - interpolation

- Interpolation bilinéaire des normales le long des cotés et de la ligne de scan



- Calcul de la couleur pour chaque pixel

Comparaison



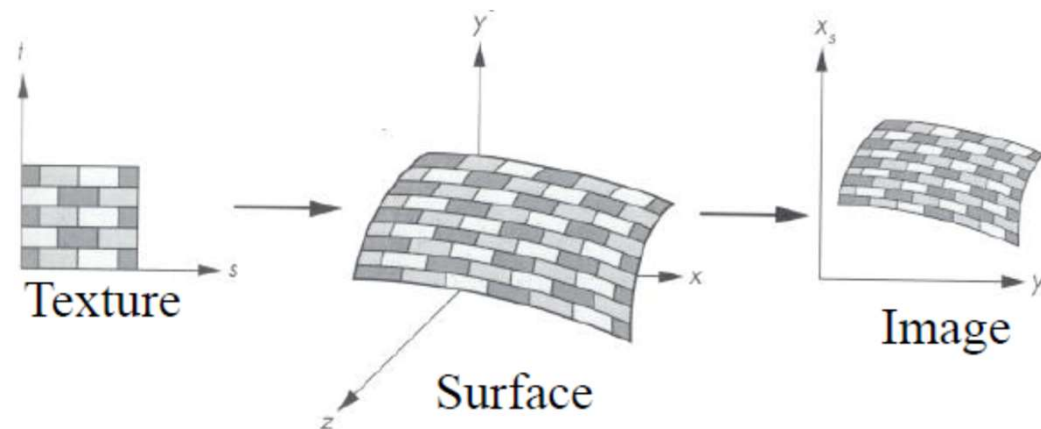
Texture mapping

- Nous savons comment “colorier” de façon lisse un polygone en conservant les subtilités liées à l’éclairage
- Comment considérer les petits détails ?
 - Solution 1 : ajouter des polygones pour chaque détail. Cette solution n’est pas viable. Cela ajoute trop de complexité à la scène.
 - Solution 2 : Garder un polygone de grande taille et utiliser une image pour représenter les détails.
- Texture mapping = utilisation d’images plaquées sur les polygones



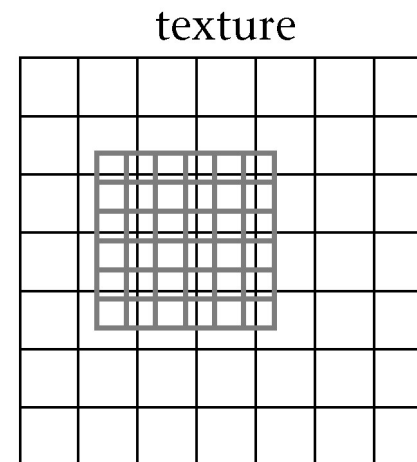
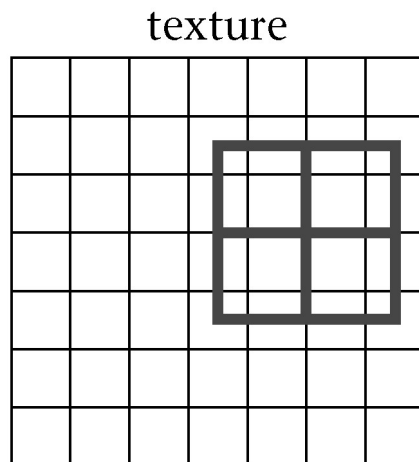
Texture Mapping

- Pourquoi utiliser des textures ?
 - Approximation des couleurs des surfaces
 - Compactage efficace des détails
- Le placage de textures standard correspond à une modification des valeurs diffuses le long du polygone
- Une **texture** est un tableau 2D de **texels** (texture elements) stockant des valeurs RGB or RGBA (Red, Green, Blue, Alpha)



Difference entre pixels et texels

- Il y a forcément une différence de correspondance et de taille entre les pixels de la fenêtre d'affichage et les texels d'une texture

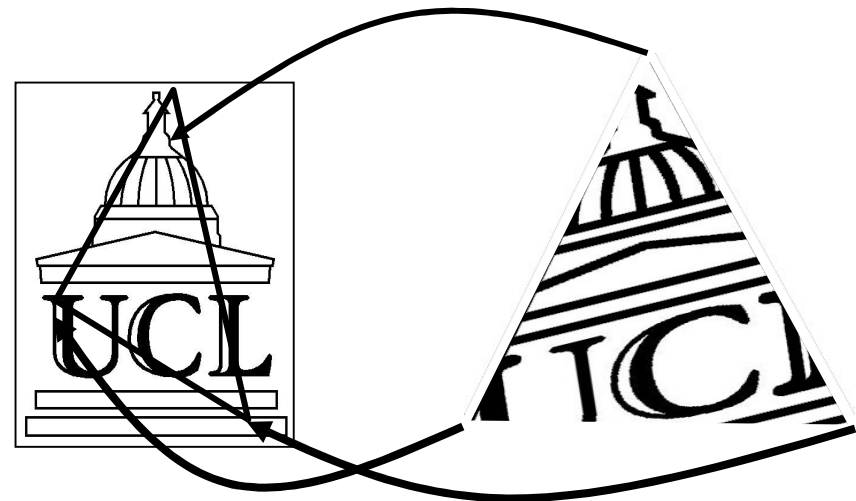
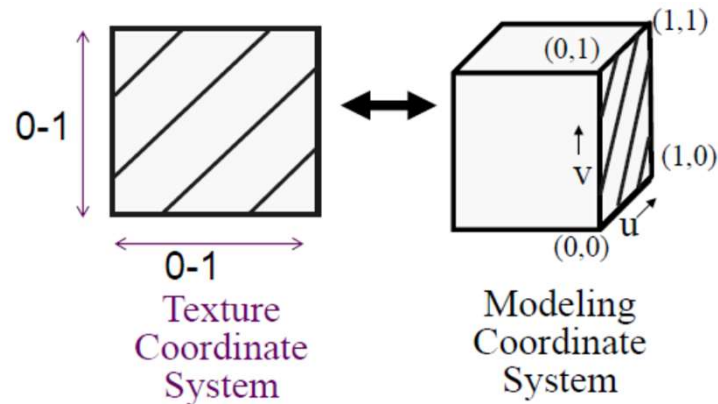


Les différentes méthodes et applications

- Texture mapping
 - Inverse Mapping
 - Bilinear interpolation
 - Perspective correction
 - Mipmapping
- Applications
 - Environment mapping
 - Bump mapping
 - Illumination mapping
 - Modulation textures
 - Image-based rendering
 -

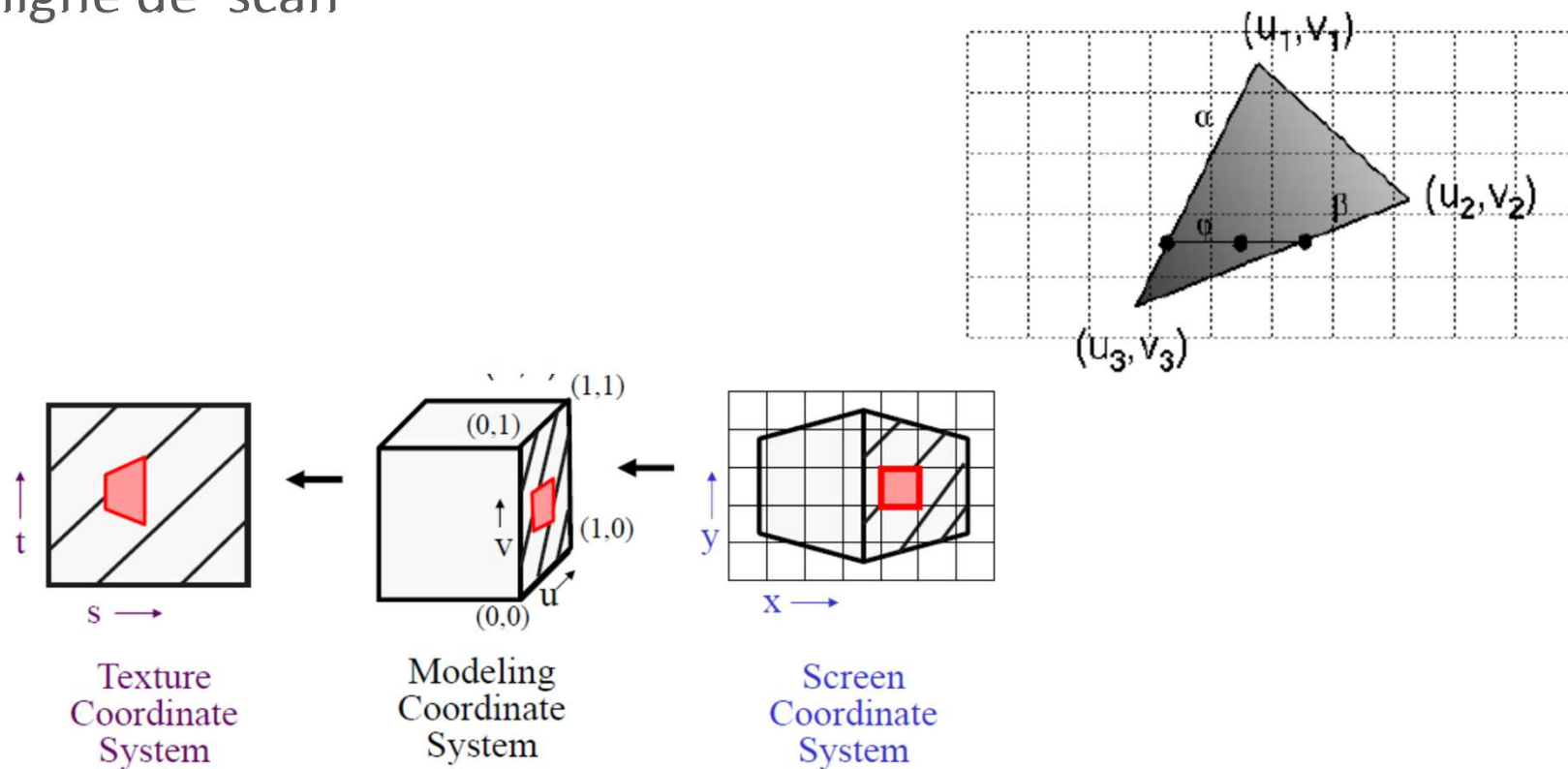
Inverse Texture Mapping

- A chaque sommet du polygone, on associe un point dans l'image, en donnant ses coordonnées (u,v) dans l'image



Inverse Texture Mapping

- A l'affichage, récupérer les valeurs des texels le long de la ligne de scan



Problèmes

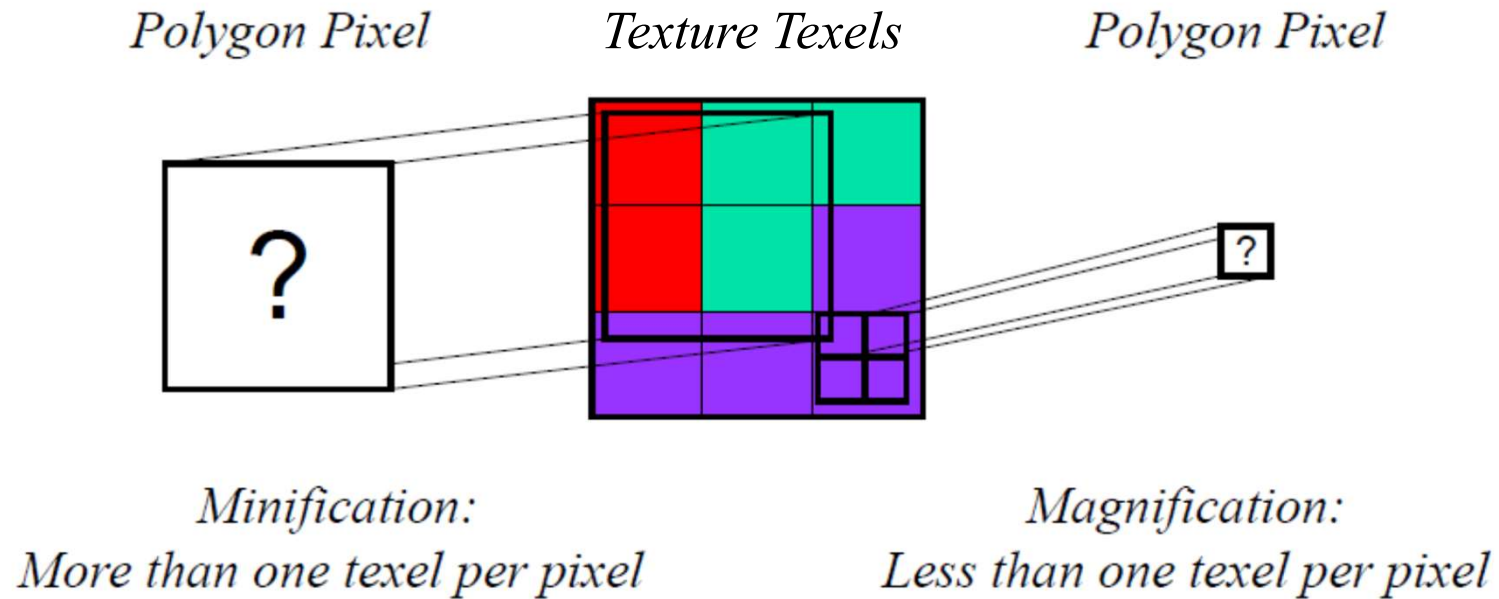
- Liés à la perspective
- Il est possible de la corriger lors de l'interpolation
 - Interpolation selon $1/z$



Echantillonnage

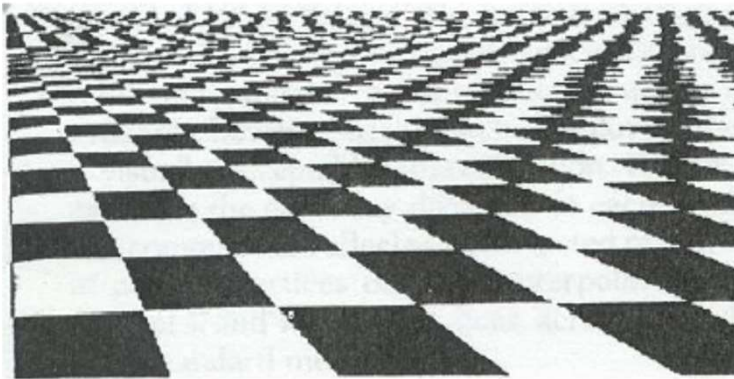
- Un pixel s'applique sur une region non rectangulaire
- En general, on n'applique le placage qu'en considérant le centre du pixel
- Il reste le problème de difference entre les texels et les pixels
 - Over-sampling : un même texel s'applique sur plusieurs pixels
 - Under-sampling : les pixels recouvrent grossièrement l'échantillonnage des texels

Minification vs. magnification

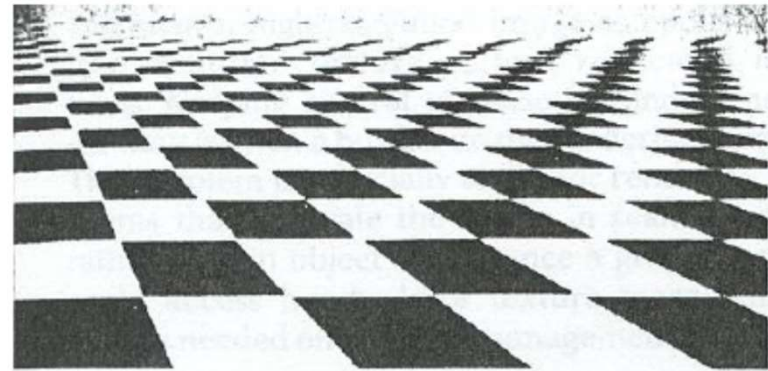


Aliasing

- Nécessité d'appliquer un filtrage



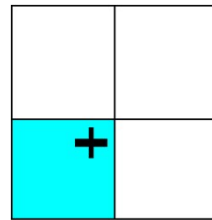
Point sampling



After area filtering

Magnification : filtrage

1. Voisinage le plus près (Nearest neighbor)

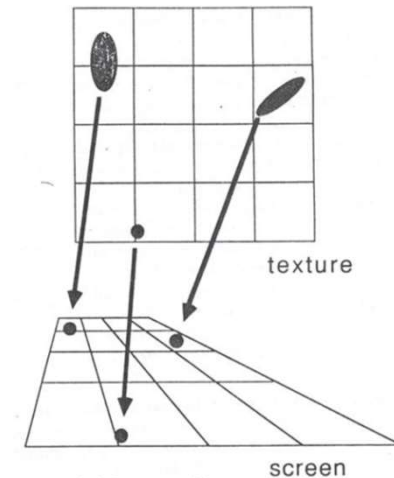
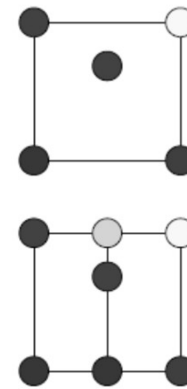


- Sélectionner le pixel dont le centre est le plus proche

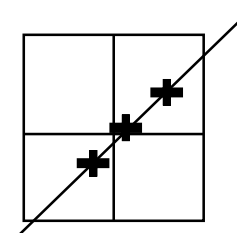
2. Bilinear



- Weighted average based on distance to pixel centre



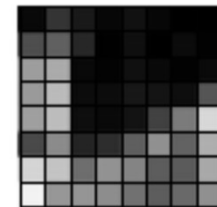
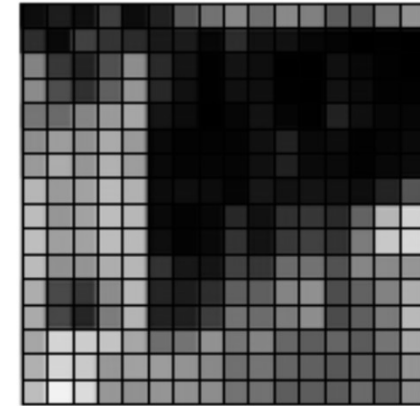
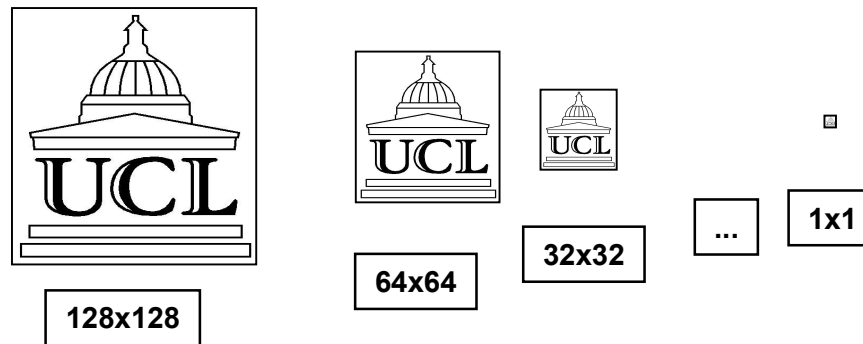
- Solutionne en partie le problème d'over-sampling en permettant un lissage entre les pixels



scanline

Mip-Mapping

- Solution pour l' under-sampling (minification) : mip-mapping
- Créer des textures de résolution différentes
- Création d'une pyramide de textures
- Possibilité d'interpoler les textures de deux couches adjacentes



Mip-mapping

- Utiliser la texture la plus proche de la résolution de l'écran ou choisir deux couches tels que les écarts entre du et dv et dx et dy soient minimisés
- Interpoler entre deux couches de textures proches
 - Interpoler entre 4 pixels pour la couche de résolution supérieure et un pixel pour la couche de plus basse résolution.



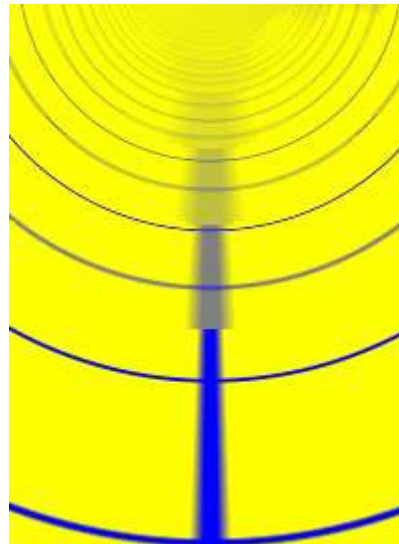
Exemples



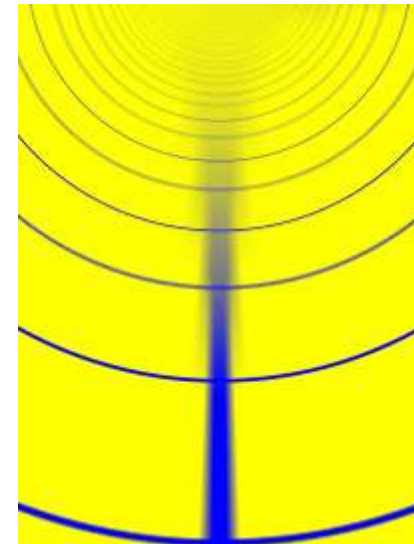
Nearest Neighbor



Bilinear Filtering

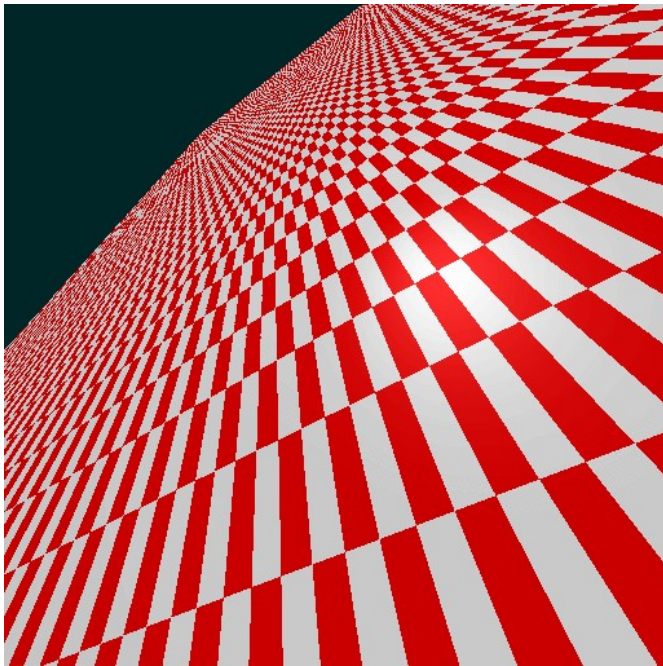


Bilinear Filtering
(distinct mip map
levels)

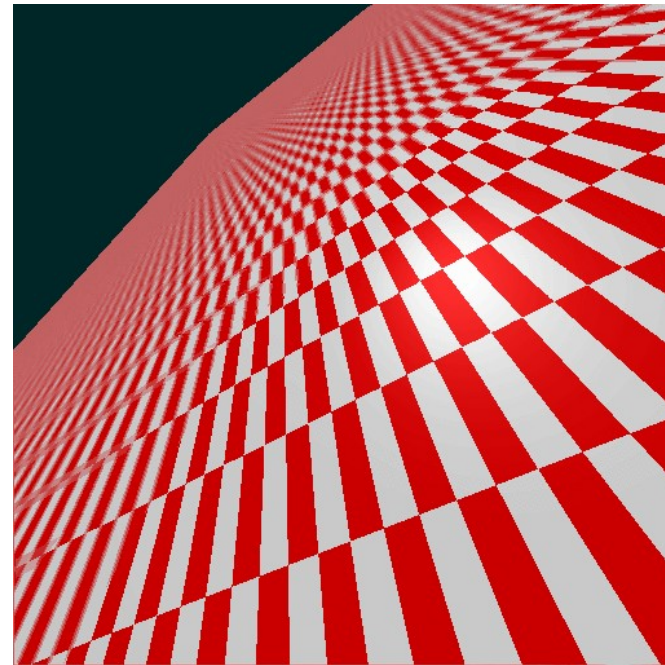


Trilinear Filtering
(Mip mapping)

Exemples



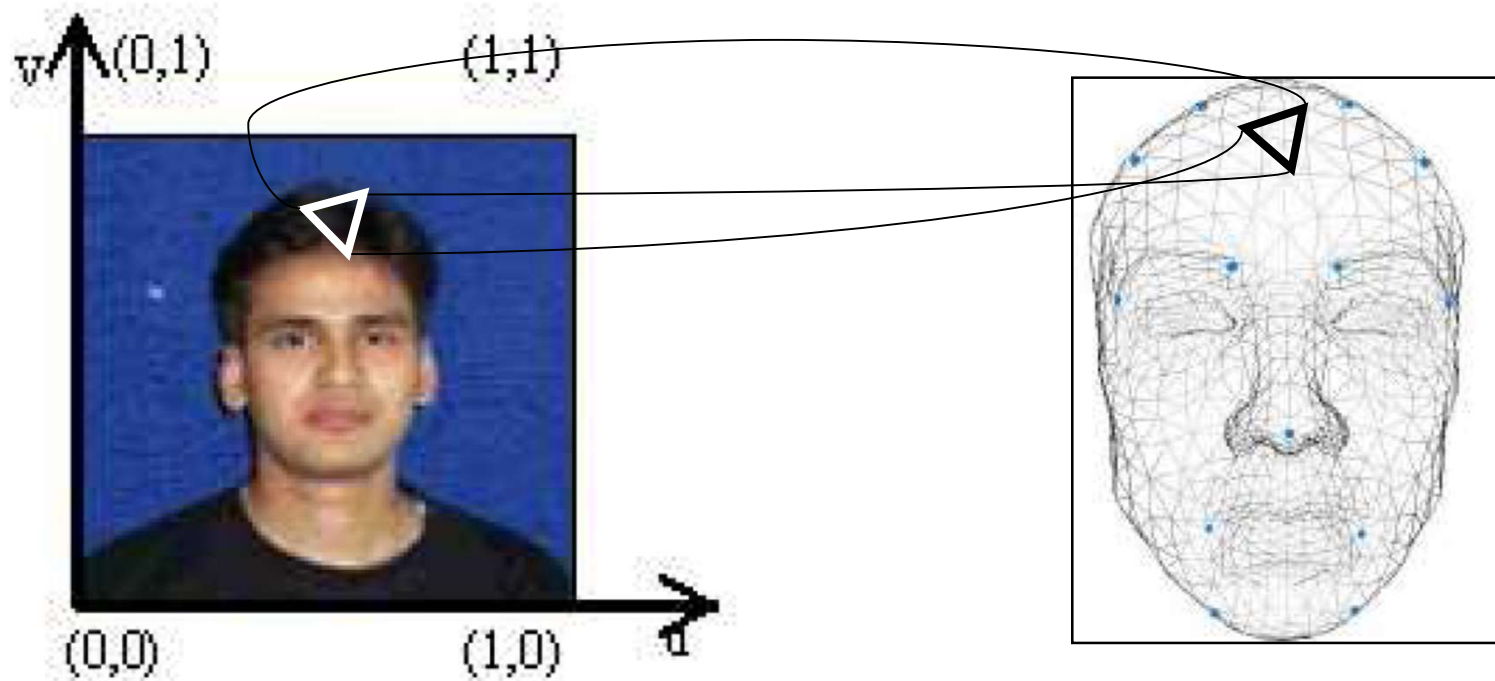
Nearest Neighbor



Mip mapping

Autres considérations

- Paramétrisation :
- Trouver les coordonnées de texture est difficile mais essentiel



Autres considérations

- Paramétrisation

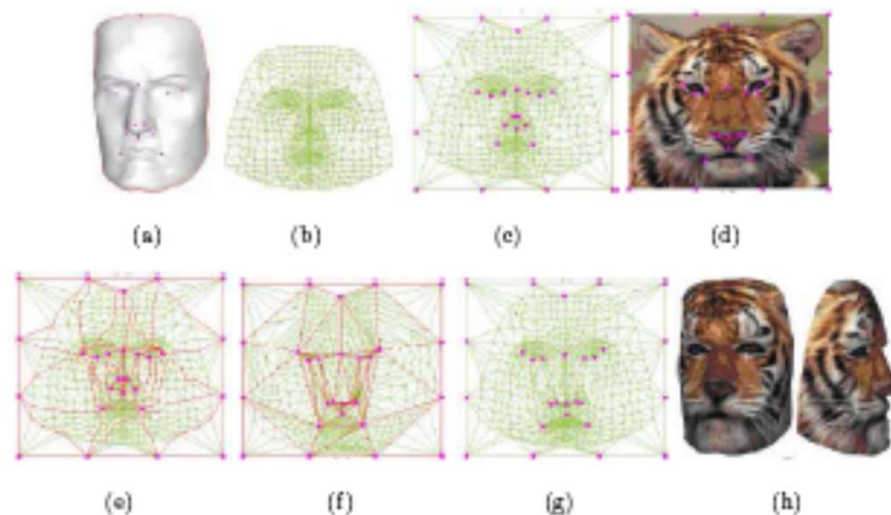


Figure 2.4: Match maker algorithm Example: Mapping a tiger face onto a human. (a) 3D model with feature vertices (b) plan input mesh (c) Parameterization with virtual boundary. (d) Texture with feature points. (e) Matching triangulation. (f) Triangle embedding. (g) Mesh after constrained smoothing. (h) Resulting textured model. Images courtesy of Kraevoy et al. [10]

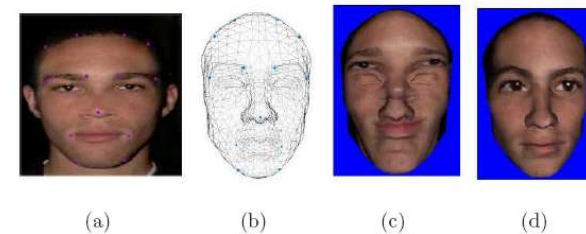
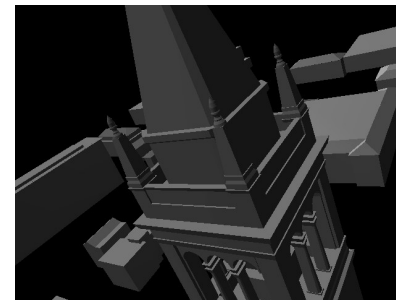
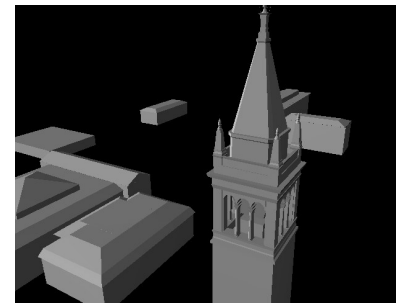
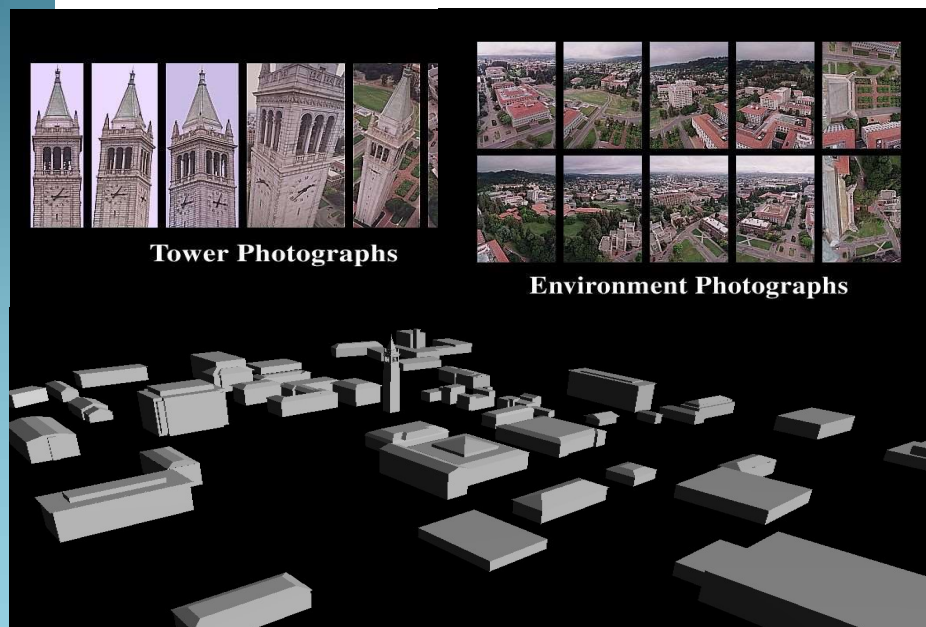


Figure 2.2: Texture mapping without constraints. (a) Texture map with feature points marked. (b) 3D face model with features corresponding to those of (a). (c) Rendering of result without any constraints. The result is not what we expected. (d) Rendering of result when constraints are satisfied using techniques in [7]. Images courtesy of Eckstein et al. [7]

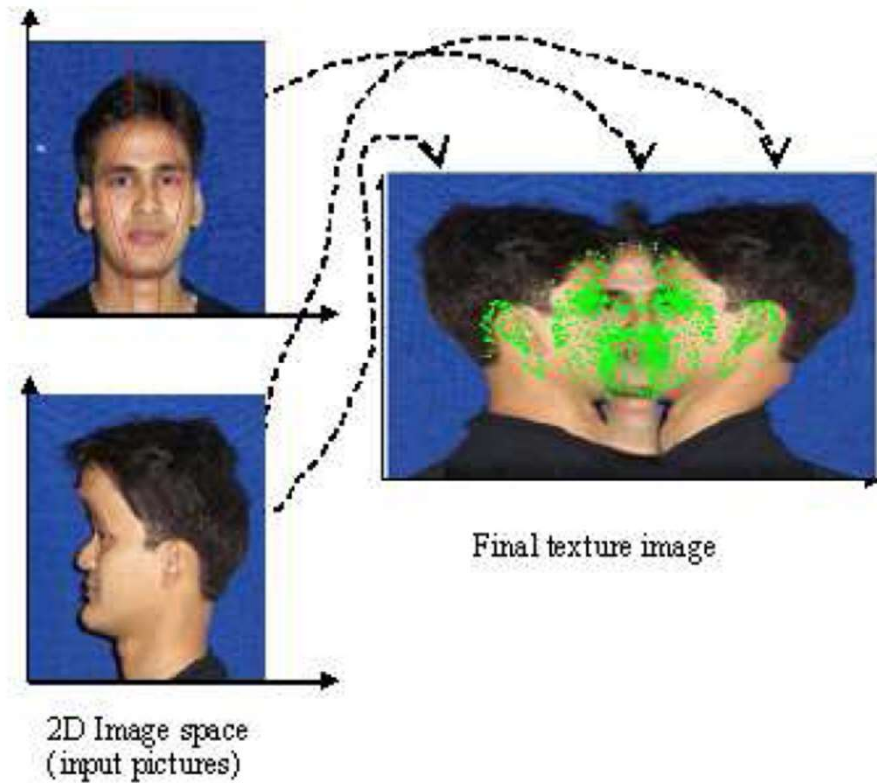
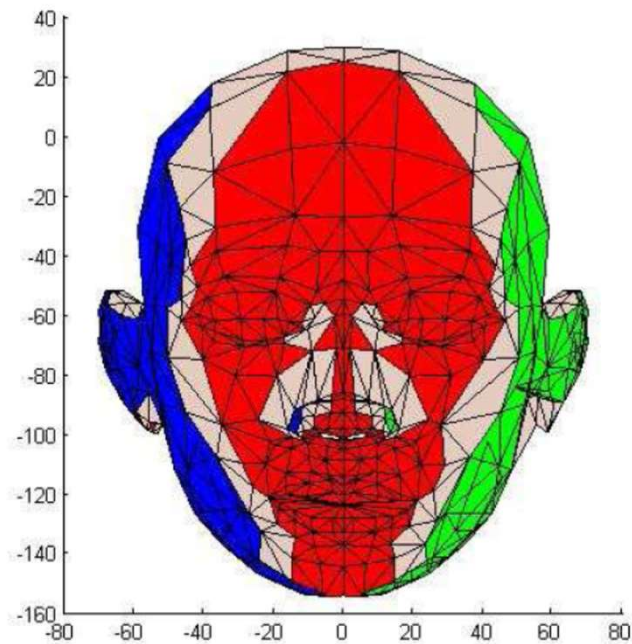
Autres considérations

- Textures provenant de plusieurs points de vue
 - Trouver les coordonnées pour gérer les occultations et les problèmes de déformations dues à la perspective



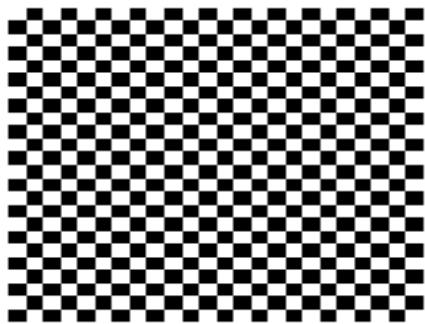
Autres considérations

- Textures provenant de plusieurs points de vue
 - Blending (mixage)

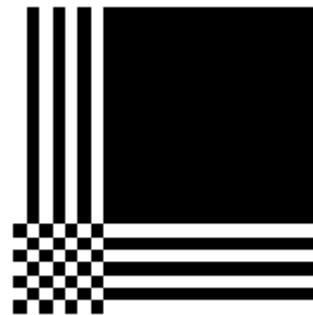


Autres considérations

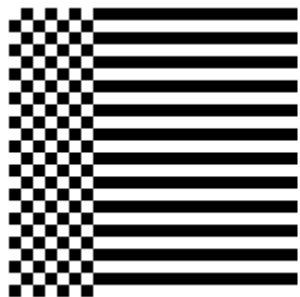
- Remplissage lorsque la texture est petite par rapport à la taille du polygone



Repeat



Clamp



Repeat + Clamp



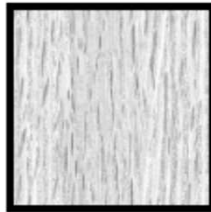
Applications

- Modulation textures : utiliser les textures en facteur multiplicateur

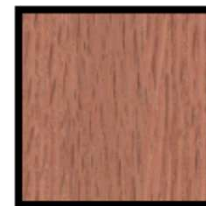
$$I = T(s, t)(I_E + K_A I_A + \sum_L (K_D (N \cdot L) + K_S (V \cdot R)^n) S_L I_L + K_T I_T + K_S I_S)$$

Texture
value

Wood texture

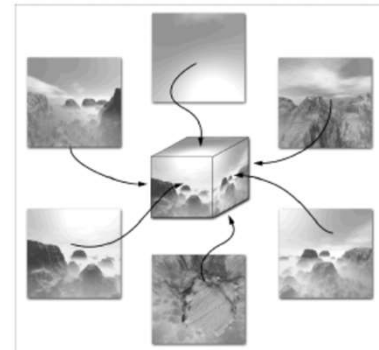


Color Value



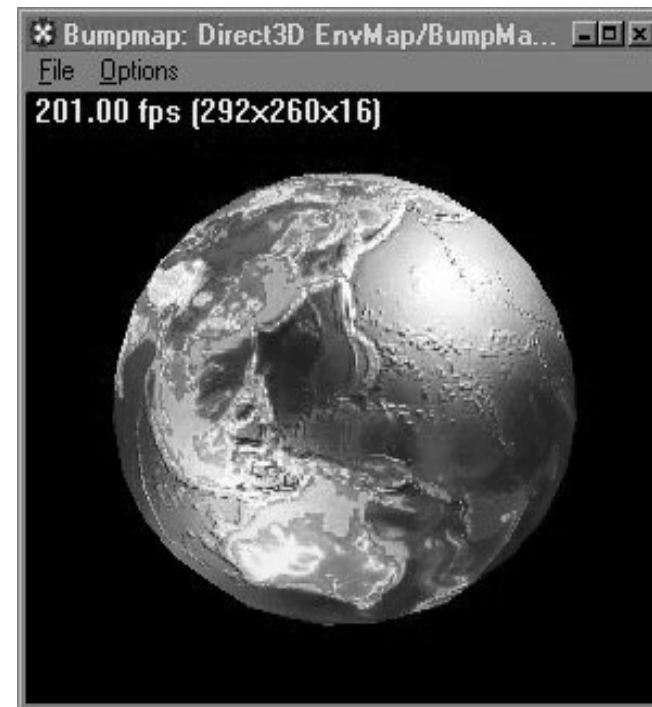
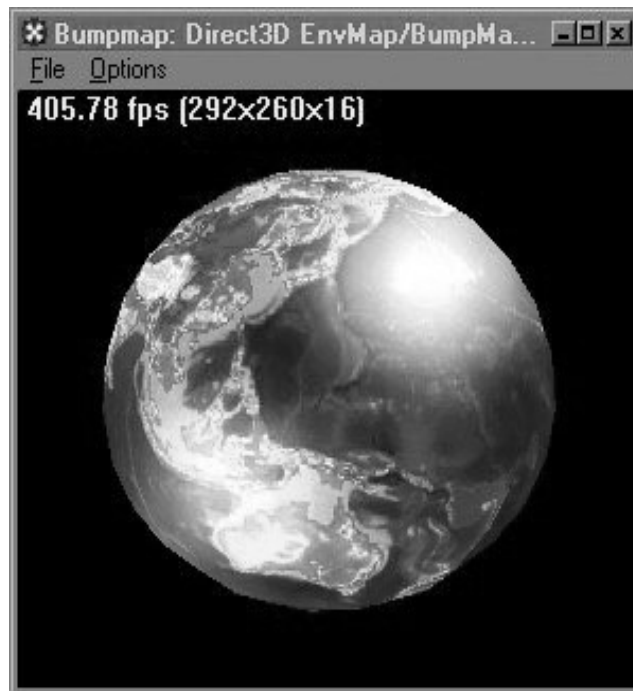
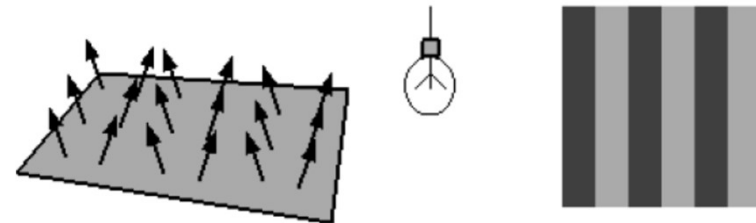
Applications

- Environnement mapping
- Modifie la couleur en jouant sur la normale
- Crée des réflexions
- La carte d'environnement est une texture de l'arrière plan



Applications

- Bump mapping : utilise une carte pour perturber les normales de l'éclairage spéculaire



Conclusion

- Solutions de rendu dans le plan image
 - Gouraud shading
 - Phong shading
- Possibilité d'ajouter des détails par l'utilisation des textures
 - Nécessite de bien trouver les coordonnées de texture et de gérer les problèmes d'échantillonnage