

## 8.Textures

### 8.1 Introduction

Textures : détail de la surface d'un objet

Augmente le réalisme de l'image par

- ajout de motif à un objet (ex : image sur une couverture d'un livre)
- répétition de motifs (ex : papier peint)
- aspect de certaines matières (rugosité)

Peu de traitements, beaucoup d'effets

1

### Introduction

- fonction de mappage (mapping) : adjonction d'un motif à une surface  
 mapping de texture 2D(carrelage, papier peint...)  
 mapping de texture solide (bois, marbre... )

Change la couleur pour un pixel donné (remplace ou module la composante ambiante et diffuse)

- fonction de perturbation (bump mapping) :  
 adjonction de rugosité à une surface lisse (granit, ...)

2

### 8.2 mapping 2D

Objectif :

plaquer une image 2D (bitmap) sur un objet 3D

Principe :

- trouver la correspondance entre l'image 2D et l'objet 3D (pas évidente)
- afficher l'objet "recouvert" de l'image (aliasing)



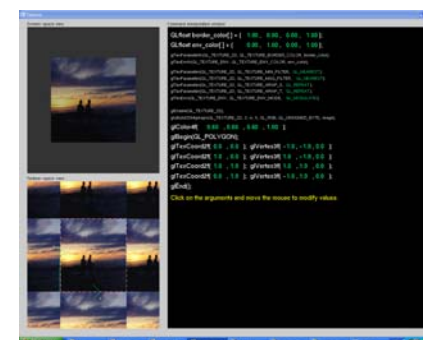
3

### mapping 2D

Correspondance entre objet 3D et bitmap :

Soit  $S(x,y,z)$  un point de l'objet 3D : calcul de  $I(u,v)$  du bitmap associé à  $S$

- pour certains objets de forme géométrique simple : association possible



4

## mapping 2D

Correspondance entre objet 3D et bitmap :

Soit  $S(x,y,z)$  : calcul de  $I(u,v)$  du bitmap associé à  $S$

pour un objet quelconque : relation difficile voire impossible à calculer  
pour certains objets de forme géométrique simple : association possible

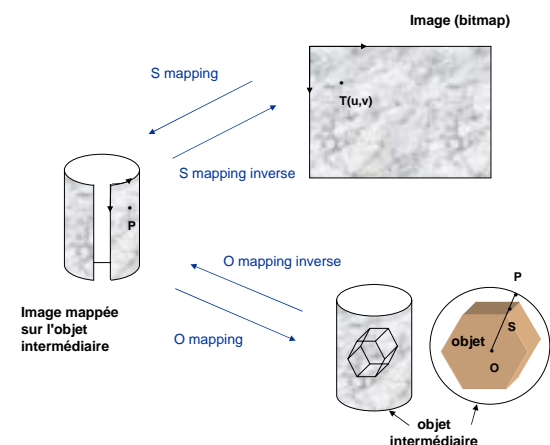
décomposition en 2 étapes

- mapper la texture 2D sur une surface 3D simple : S mapping  
 $T(u,v) \rightarrow P(x_i, y_i, z_i)$
- mapper le modèle de la surface 3D sur l'objet : O mapping  
 $P(x_i, y_i, z_i) \rightarrow S(x_w, y_w, z_w)$

Application : mapping inverse

5

## mapping 2D



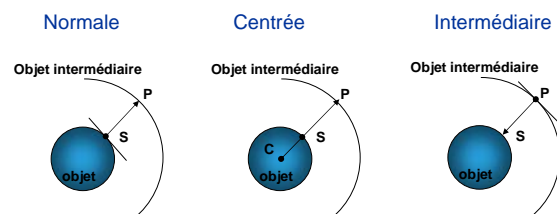
6

## mapping 2D

Objets intermédiaires

plan cube cylindre sphère

Correspondance entre objet 3D et objet intermédiaire:



7

## mapping 2D

Le mapping sphérique centré

- Le O mapping inverse

Correspondance entre un point de l'objet 3D et l'objet intermédiaire (sphère)

Sphère centrée sur le centre de gravité de l'objet 3D

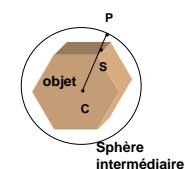
sphère englobe totalement l'objet 3D.

Calcul de l'intersection de la droite CS avec la sphère

$$X_p = X_c + (X_s - X_c)\rho$$

$$Y_p = Y_c + (Y_s - Y_c)\rho$$

$$Z_p = Z_c + (Z_s - Z_c)\rho$$



8

## mapping 2D

- Le S mapping

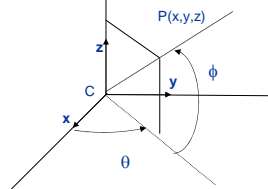
Trouver le point T(u,v) de la texture correspondant au point P de la sphère intermédiaire

Soient les coordonnées sphériques ( $\theta, \phi$ ) du point P (dans le repère lié à la sphère)

$$P' = P - C$$

$$\theta = \arctan\left(\frac{Y_{P'}}{X_{P'}}\right)$$

$$\phi = \arctan\left(\frac{Z_{P'}}{\sqrt{X_{P'}^2 + Y_{P'}^2}}\right)$$



9

## mapping 2D

Calcul des coordonnées u,v de la texture

$$u = \frac{\theta}{2\pi} + \frac{1}{2} \quad -\pi \leq \theta \leq \pi$$

$$v = -\frac{\phi}{\pi} + \frac{1}{2} \quad -\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$$

Normalisation pour des valeurs comprises entre 0 et 1

signe - car centre du repère associé au bitmap en haut à gauche et v vers le bas

10

## mapping 2D

### Le mapping cylindrique "centré"

- Le O mapping inverse

Correspondance entre un point de l'objet 3D et l'objet intermédiaire (cylindre)

cylindre centré sur le centre de gravité de l'objet 3D (centre du cylindre)

cylindre englobe totalement l'objet 3D (rayon R et hauteur H du cylindre)

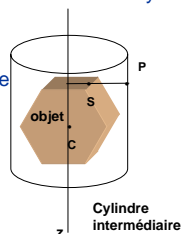
cylindre parallèle à l'axe des z

Calcul des coordonnées du point P sur le cylindre

$$X_p = X_c + (X_s - X_c)p$$

$$Y_p = Y_c + (Y_s - Y_c)p$$

$$Z_p = Z_s$$



11

## mapping 2D

Calcul des coordonnées u,v de la texture

$$u = \frac{\theta}{2\pi} + \frac{1}{2} \quad -\pi \leq \theta \leq \pi$$

$$v = 1 - \frac{Z_p - Z_{\min}}{H}$$

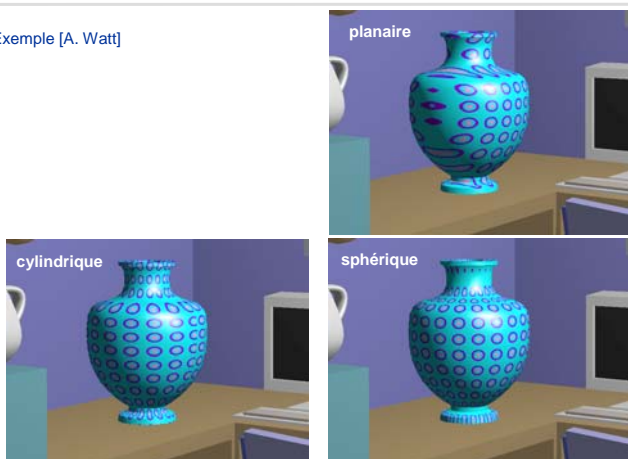
Normalisation pour des valeurs comprises entre 0 et 1

signe - car centre du repère associé au bitmap en haut à gauche et v vers le bas

12

## mapping 2D

Exemple [A. Watt]



13

## mapping 2D

Les limites du mapping 2D

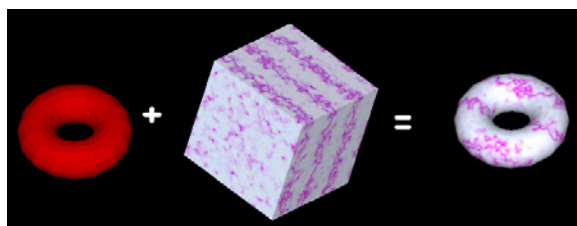
bien défini dans le cas d'objets simples  
déformations de la texture  
effets de pixelisation si la taille du bitmap est trop petite par rapport à la taille de la surface  
problèmes de discontinuités dans le cas de certaines textures



14

## 8.3 les textures solides

- la texture et la forme d'un objet sont complètement séparées
- l'objet semble *sculpté* dans le matériau



15

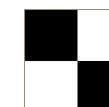
## Textures solides

Principe

Cette méthode consiste à remplacer la représentation explicite de la texture (image 2D ou même 3D) par une fonction mathématique variant dans les trois dimensions de l'espace. La texture n'est donc plus contenue dans une structure de données mais elle est décrite par un programme (texture par fonction ou texture procédurale)

Exemple : texture de damier noir et blanc.

Mapping 2D : répéter le motif suivant



16

## Textures solides

### texture procédurale :

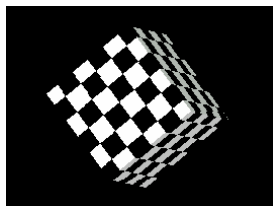
Début

```
a = [ PartieEntière(x) +PartieEntière(y) +
      partieEntière(z) ] Modulo 2
```

```
Si a=0 Alors retourner la couleur blanche
Sinon retourner la couleur noire
```

Fin

Cet algorithme prend en paramètre les coordonnées (x,y,z) d'un point à la surface d'un objet et retourne la couleur de la case dans laquelle il se trouve.



17

## Textures solides

### avantages

- représentation compacte
- pas de résolution fixe
- domaine de définition infini
- peut être paramétrée (classe de textures)

### inconvénients

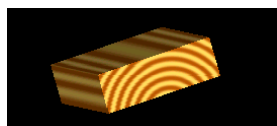
- difficile à définir
- difficile à prédire et à contrôler
- calcul parfois plus lent que le mapping
- antialiasing difficile

18

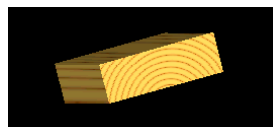
## Textures solides

### Exemples [TX A98]

Allan Watt, *Computer Graphics*.  
coordonnées cylindriques  
la variation entre 2 couleurs en sinus  
cylindres concentriques bigarrés.



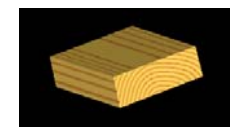
Introduction d'un biais pour renforcer la couleur claire  
Introduction d'un terme qui rendra compte de l'écartement entre les veines du bois.



19

## Textures solides

variation pour que la section de bois ne donne pas des cercles, mais des cercles légèrement déformés.



Introduction d'un terme de déphasage pour avoir une torsion radiale de sorte que les cercles paraissent moins concentriques.



torsion longitudinale pour avoir un effet plus réaliste sur la longueur de la planche.



20

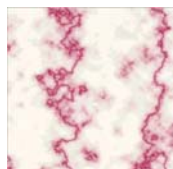
## Textures solides

### Exemples

fonction marbre

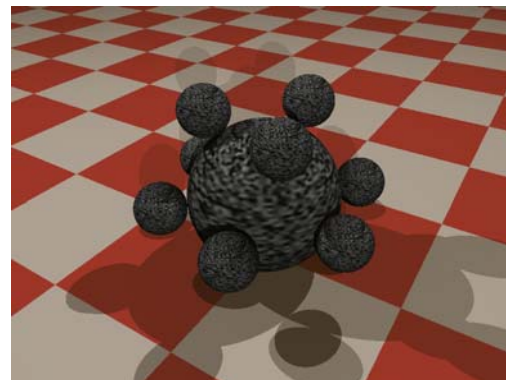


application d'une fonction  
'turbulence' sur la coordonnées x



21

## Textures solides [ G. Lecot TX01]

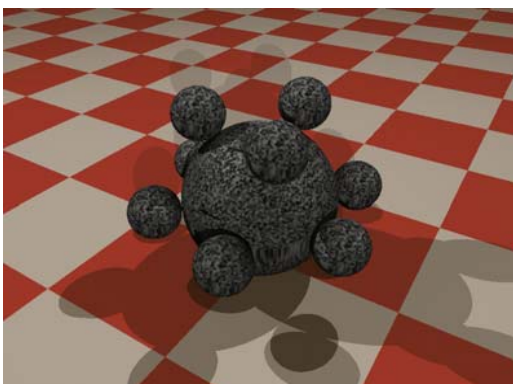


Noise 2D

22

## Textures solides

Noise 3D



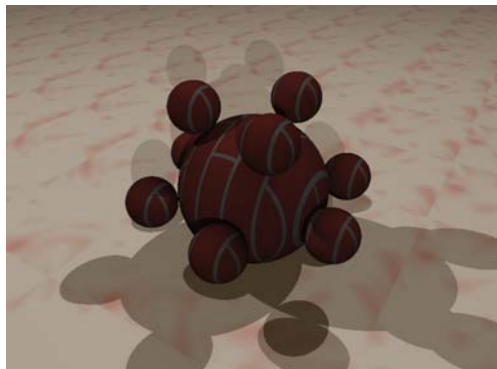
23

## Mapping de briques



Mapping 2D  
sphérique

24



Texture 3D

25

## 8. 4 Bump Mapping

Le bump mapping est une technique permettant de donner l'illusion de relief, sans augmenter la géométrie.[BLINN 78]

### Principe du bump mapping

Blinn s'est inspiré de la méthode d'ombrage de Phong:  
utilisation des normales.

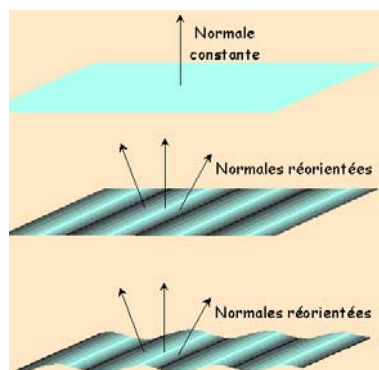
Un vecteur normal est toujours perpendiculaire à la surface,

Si on incline une normale, cela signifie qu'à l'endroit où la normale est inclinée, la surface l'est aussi.

26

## Bump Mapping

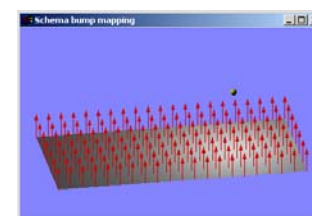
Utilisation d'une surface  
et d'un modèle d'éclairage



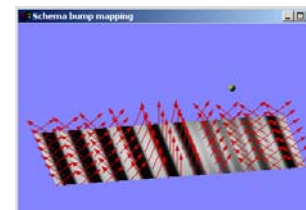
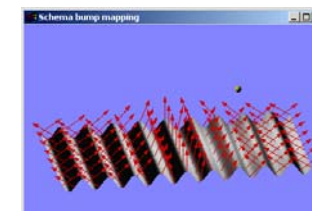
Nicolas JANEY/ univ Franche Comté

27

## Bump Mapping

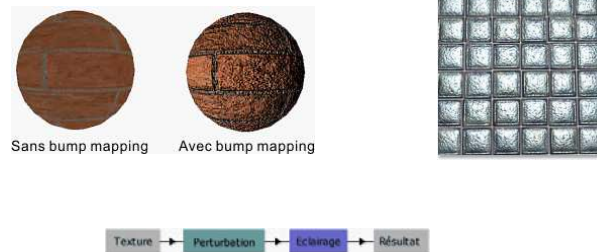


Nicolas JANEY/ univ Franche Comté



28

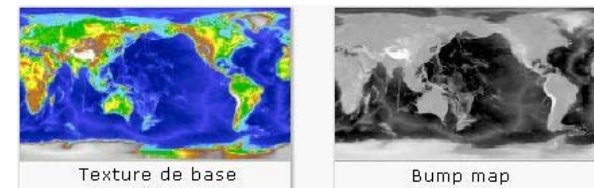
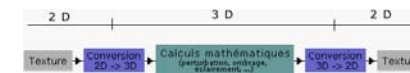
## Bump Mapping (exemples)



29

## Bump Mapping

On peut utiliser une texture pour stocker les normales (ou toute autre information servant à recalculer les normales) dans une texture : c'est la Bump Map



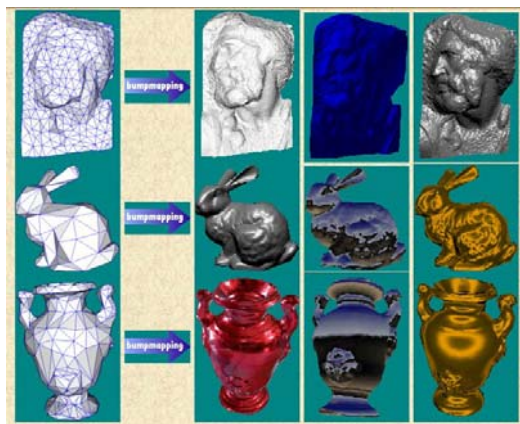
information de relief

<http://www.university.com/> J.F. Maquine

30

## Bump mapping

<http://vcg.iei.pi.cnr.it>



31

## 8.5 Textures et OpenGL : mapping 2D

### 1. définition de la texture

- Tableau de pixels
- Importation d'une image : .bmp, .jpeg (utiliser des fonctions appropriées)

`GLubyte Texture[16] =`

```
{
    0,0,0,0, 0xFF,0xFF,0xFF,0xFF,
    0xFF,0xFF,0xFF,0xFF, 0,0,0,0
};
```

//Image (2x2)

### 2. activation du mode texture et création de la texture

32



## Textures et OpenGL

```
void InitGL()
{
    glClearColor(.5,.5,.5,0);           //Change la couleur du fond
    glEnable(GL_DEPTH_TEST);            //Active le depth test
    glEnable(GL_TEXTURE_2D);            //Active le texturing
    glGenTextures(1,&Nom);               //Génère un n° de texture
    glBindTexture(GL_TEXTURE_2D,Nom);    //Sélectionne ce n°
    glTexImage2D (
        GL_TEXTURE_2D,                  //Type : texture 2D
        0,                              //Mipmap : aucun
        4,                              //Couleurs : 4
        2,                              //Largeur : 2
        2,                              //Hauteur : 2
        0,                              //Largeur du bord : 0
        GL_RGBA,                        //Format : RGBA
        GL_UNSIGNED_BYTE,               //Type des couleurs
        Texture                          //Adresse de l'image
    );
```

33

## Textures et OpenGL

```
glTexParameteri(GL_TEXTURE_2D,        //Sans Filtres
                 GL_TEXTURE_MIN_FILTER,
                 GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D,
                 GL_TEXTURE_MAG_FILTER,
                 GL_NEAREST);
}
```

34

## Textures et OpenGL

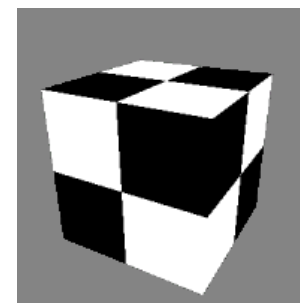
### 3. Coordonnées de la texture

```
glBegin(GL_QUADS
glTexCoord2i(0,0);    glVertex3i(-1,-1,-1);
glTexCoord2i(1,0);    glVertex3i(+1,-1,-1);
glTexCoord2i(1,1);    glVertex3i(+1,+1,-1);
glTexCoord2i(0,1);    glVertex3i(-1,+1,-1);    //1 face
glTexCoord2i(0,0);    glVertex3i(-1,-1,+1);
glTexCoord2i(1,0);    glVertex3i(+1,-1,+1);
glTexCoord2i(1,1);    glVertex3i(+1,+1,+1);
glTexCoord2i(0,1);    glVertex3i(-1,+1,+1);    //2 faces
...
glEnd();
```

35

## Textures et OpenGL

[antoche@altern.org](mailto:antoche@altern.org)

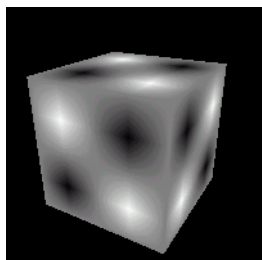


```
glHint( GL_PERSPECTIVE_CORRECTION_HINT,
        GL_NICEST);
```

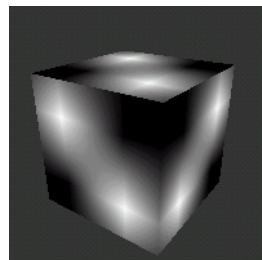
36

## Textures et OpenGL

antoche@altern.org



```
glTexParameteri(GL_TEXTURE_2D,  
GL_TEXTURE_MIN_FILTER,  
GL_LINEAR);  
glTexParameteri(GL_TEXTURE_2D,  
GL_TEXTURE_MAG_FILTER,  
GL_LINEAR);
```



```
glTexParameteri(GL_TEXTURE_2D,  
GL_TEXTURE_WRAP_S,  
GL_REPEAT);  
glTexParameteri(GL_TEXTURE_2D,  
GL_TEXTURE_WRAP_T,  
GL_REPEAT);
```

37

## 8.6 Applications

- Billboarding
  - Visualisation d'objets complexes non modélisés (arbres)
  - Utilisation de plans
- Mapping d'environnement
  - Limites d'une scène (ciel)
  - Cube ou sphérique
  - Réflexion des objets sur les autres
- Texture d'ombres
  - Calcul des ombres et mémorisation dans une texture
- LOD (level of detail)

38

## Billboarding

« Billboard » : panneau de bord de route

Dessin d'objets en 2D mais qui apparaissent comme des objets en 3D  
On applique une texture contenant l'image de l'objet sur une primitive (face) rectangulaire. Cette face est tournée pour qu'elle soit toujours en face de l'observateur

Des parties du « billboard » peuvent être transparentes

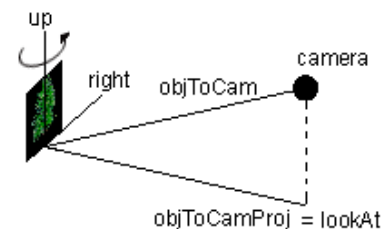
Objets symétriques



39

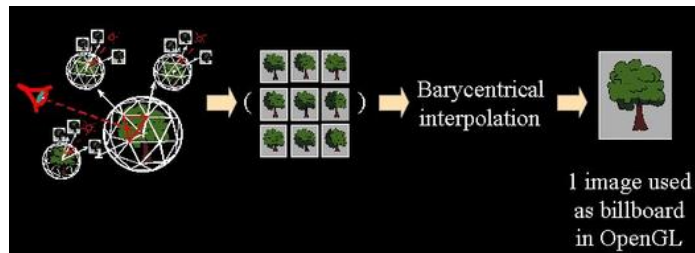
## Billboarding

antonio.ramires@lighthouse3d.com



40

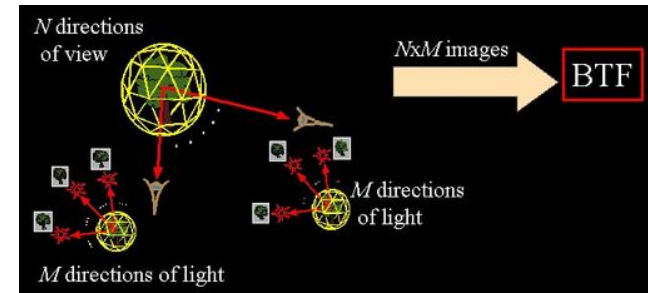
## Billboarding



Alexandre.Meyer@imag.fr

41

## Billboarding



42

## Billboarding

### Exemple



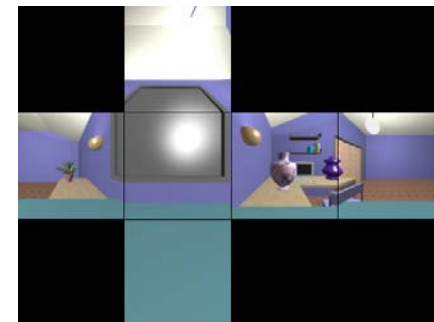
43

## Mapping d'environnement

Réflexion des objets sur les autres:

Le point de vue est placé au centre de l'objet  
L'objet est enlevé de la scène  
Calcul de 6 vues  
Assemblage (cube)

Mapping sur l'objet



Exemple (A. Watt)  
Résolution de chaque image 32\*32

44

## Mapping d'environnement

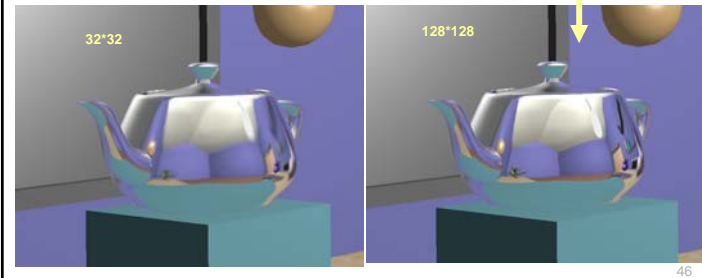
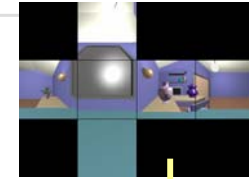
Résultat



45

## Mapping d'environnement

Changement de la résolution :  
128\*128 pour chaque face du cube



46

## Mapping d'environnement

Les « sky box »



47

## Mapping d'environnement

Exemple (tp final lo12 )



48