

Texturer un objet 3D créé au format STL

C.Turrier - 30 novembre 2020

Sommaire

1- Créer un fichier cube.stl avec openscad	3
2- Manipuler le fichier cube.stl avec meshlab	4
2.1- Importer le fichier cube.stl dans meshlab	4
2.2- Attribuer une texture au cube dans meshlab	5
2.3- Exporter le cube texturé au format obj	8
2.4- Ouvrir le fichier cube.obj avec meshlab	13
3- Tester des filtrages de cube.stl avec meshlab.....	16
3.1- Sans filtrage.....	16
3.2- Filtrage Parametrization Trivial Per-Triangle /Basic...	17
3.3- Filtrage Parametrization : Flat Plane.....	20
4- Texturer le fichier cube.obj à la main.....	22
4.1- Texturer une seule face.....	22
4.2- Texturer les 6 faces du cube à l'identique.....	24
4.3- Texturer les 6 faces du cube avec 6 textures.....	26
4.4- Texturer les 6 faces avec une texture globale.....	29

openscad est un logiciel libre de modélisation **3D** dit «paramétrique» car il permet de créer des objets 3D non pas à l'aide de la souris comme on le fait habituellement mais à l'aide d'un **langage de programmation** qui est intégré au logiciel.

Les objets **3D** ainsi créés sont associés à des **programmes sources** et peuvent être exportés dans des fichiers texte dotés de l'extension **STL** (Stereo-Lithography). Ce format, très utilisé dans le domaine des imprimantes 3D, décrit uniquement la **forme** des objets (**sommets** organisés en **faces** visibles de **triangles**) et non pas leur **texture**.

En effet, avec le format **STL**, les objets **3D** ne peuvent pas être dotés d'une **texture**. Il peuvent être dotés d'une **couleur** mais celle-ci est uniquement visible dans openscad. Elle n'est pas visible si on importe le fichier .stl avec un autre logiciel de modélisation ou de visualisation 3D. En effet, ces logiciels n'importent en fait que la **géométrie** (c'est à dire la forme) de l'objet stl et non pas les couleurs éventuellement indiquées dans les fichiers sources openscad.

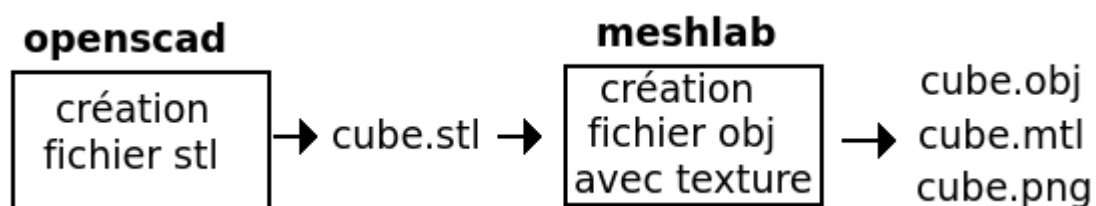
Pour pouvoir texturer un objet **3D** ayant le format **STL**, il faut donc obligatoirement exporter ce fichier dans un autre format, compatible avec le **mappage de texture**.

Le mappage de texture peut se définir comme l'établissement d'une carte de correspondance entre la surface d'un fichier objet 3D d'une part et un fichier image d'autre part.

Pour réaliser cette opération, on peut opérer en suivant les deux étapes suivantes :

- 1) Créer un objet STL avec openscad;
- 2) Importer et exporter un objet STL avec meshlab

Le fichier obtenu à l'issue de l'étape 2 (fichier exporté avec le format obj), correspond exactement au fichier stl d'origine (en terme de géométrie) mais possède en plus une texture définie dans un fichier image png.



1- Créer un fichier cube.stl avec openscad

➡ On ouvre **openscad**.

Le but recherché ici n'est pas de créer une forme **3D** compliquée mais simplement de détailler le déroulement des étapes pré-citées.

➡ On crée donc simplement un cube de 10 mm de coté.

L'instruction correspondante dans openscad est la suivante.

cube(10) ;

➡ On compile (**F5**)

➡ On restitue (**F6**) puis on exporte dans «**cube.stl**»

On obtient le fichier **cube.stl** suivant, contenant la description de $6 \times 2 = 12$ triangles pour 8 sommets.

cube.stl

solid OpenSCAD_Model		
facet normal -0 0 1 outer loop vertex 0 10 10 vertex 10 0 10 vertex 10 10 10 endloop endfacet	facet normal 0 -1 0 outer loop vertex 0 0 0 vertex 10 0 10 vertex 0 0 10 endloop endfacet	facet normal 0 1 -0 outer loop vertex 10 10 0 vertex 0 10 10 vertex 10 10 10 endloop endfacet
facet normal 0 0 1 outer loop vertex 10 0 10 vertex 0 10 10 vertex 0 0 10 endloop endfacet	facet normal 0 -1 -0 outer loop vertex 10 0 10 vertex 0 0 0 vertex 10 0 0 endloop endfacet	facet normal 0 1 0 outer loop vertex 0 10 10 vertex 10 10 0 vertex 0 10 0 endloop endfacet
facet normal 0 0 -1 outer loop vertex 0 0 0 vertex 10 10 0 vertex 10 0 0 endloop endfacet	facet normal 1 -0 0 outer loop vertex 10 0 10 vertex 10 10 0 vertex 10 10 10 endloop endfacet	facet normal -1 0 0 outer loop vertex 0 0 0 vertex 0 10 10 vertex 0 10 0 endloop endfacet
facet normal -0 0 -1 outer loop vertex 10 10 0 vertex 0 0 0 vertex 0 10 0 endloop endfacet	facet normal 1 0 0 outer loop vertex 10 10 0 vertex 10 0 10 vertex 10 0 0 endloop endfacet	facet normal -1 -0 0 outer loop vertex 0 10 10 vertex 0 0 0 vertex 0 0 10 endloop endfacet
endsolid OpenSCAD_Model		

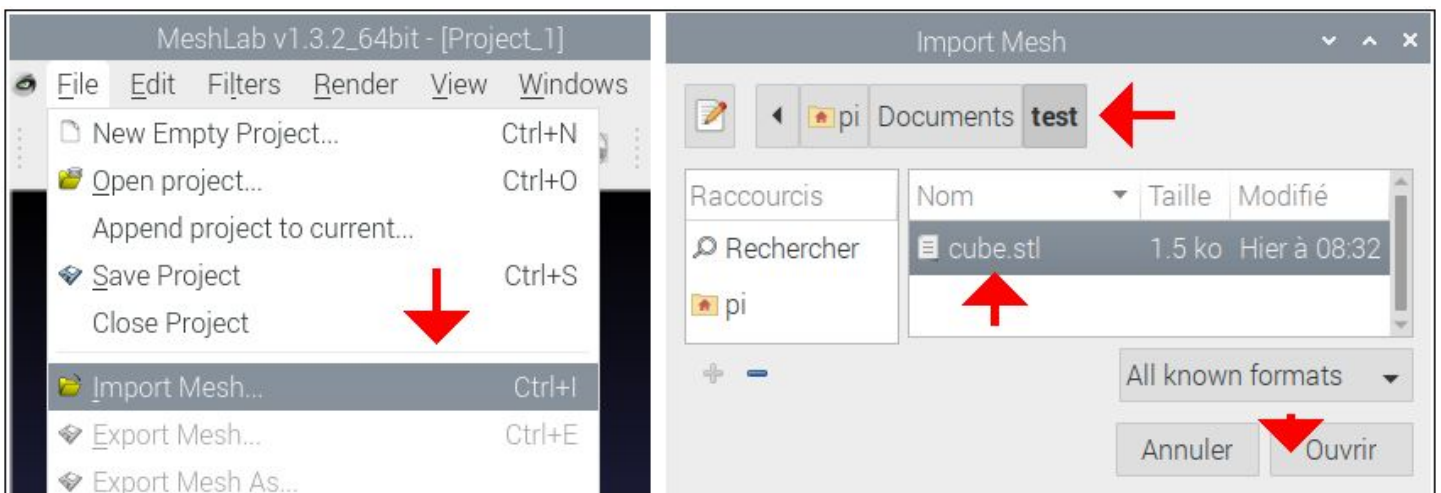
2- Manipuler le fichier cube.stl avec meshlab

2.1- Importer le fichier cube.stl dans meshlab

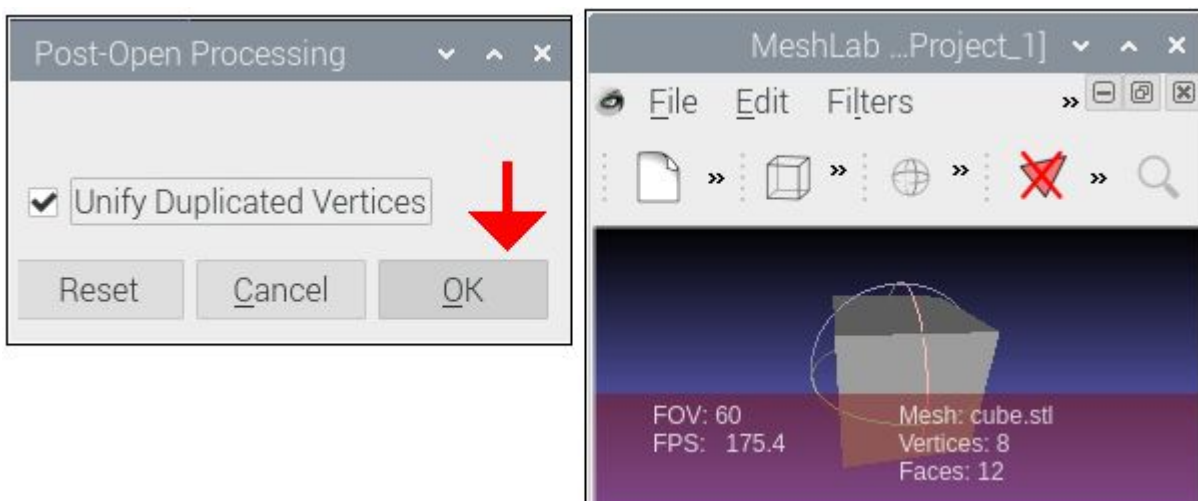
Le logiciel **openscad** ne permet pas d'exporter les fichiers au format **obj**. Pour effectuer cette opération, on utilise ici le logiciel **meshlab**.

Meshlab est un logiciel **open source** pour le traitement et l'édition de maillages triangulaires 3D (triangular meshes). Il fournit un ensemble d'outils pour éditer, nettoyer, réparer, inspecter, rendre, texturer et convertir des maillages.

➔ On ouvre **meshlab**. On clique **File/Import Mesh...** puis, dans la boîte de dialogue **Import mesh** qui s'ouvre on sélectionne le fichier **cube.stl** (créé avec openscad) et on clique le bouton **Ouvrir**.

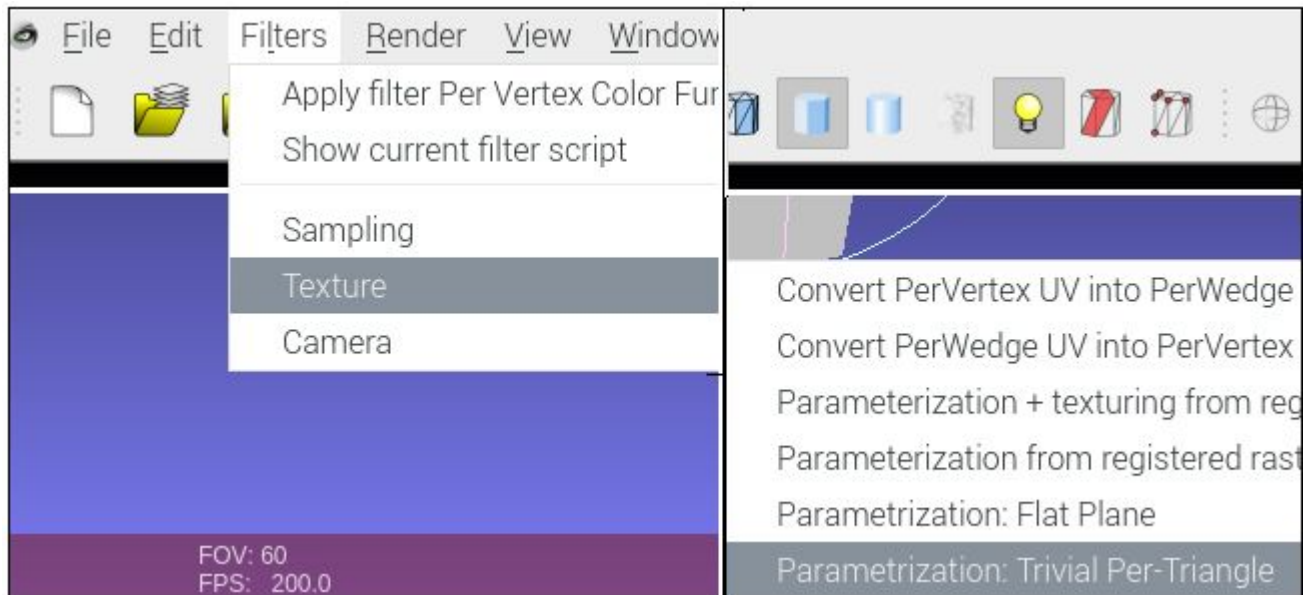


Une boîte de message **Post-Open Processing** s'ouvre automatiquement. On clique le bouton **OK**. Le cube s'affiche alors automatiquement dans la fenêtre de **MeshLab**. On peut ajuster sa taille à l'aide de la molette de la souris.

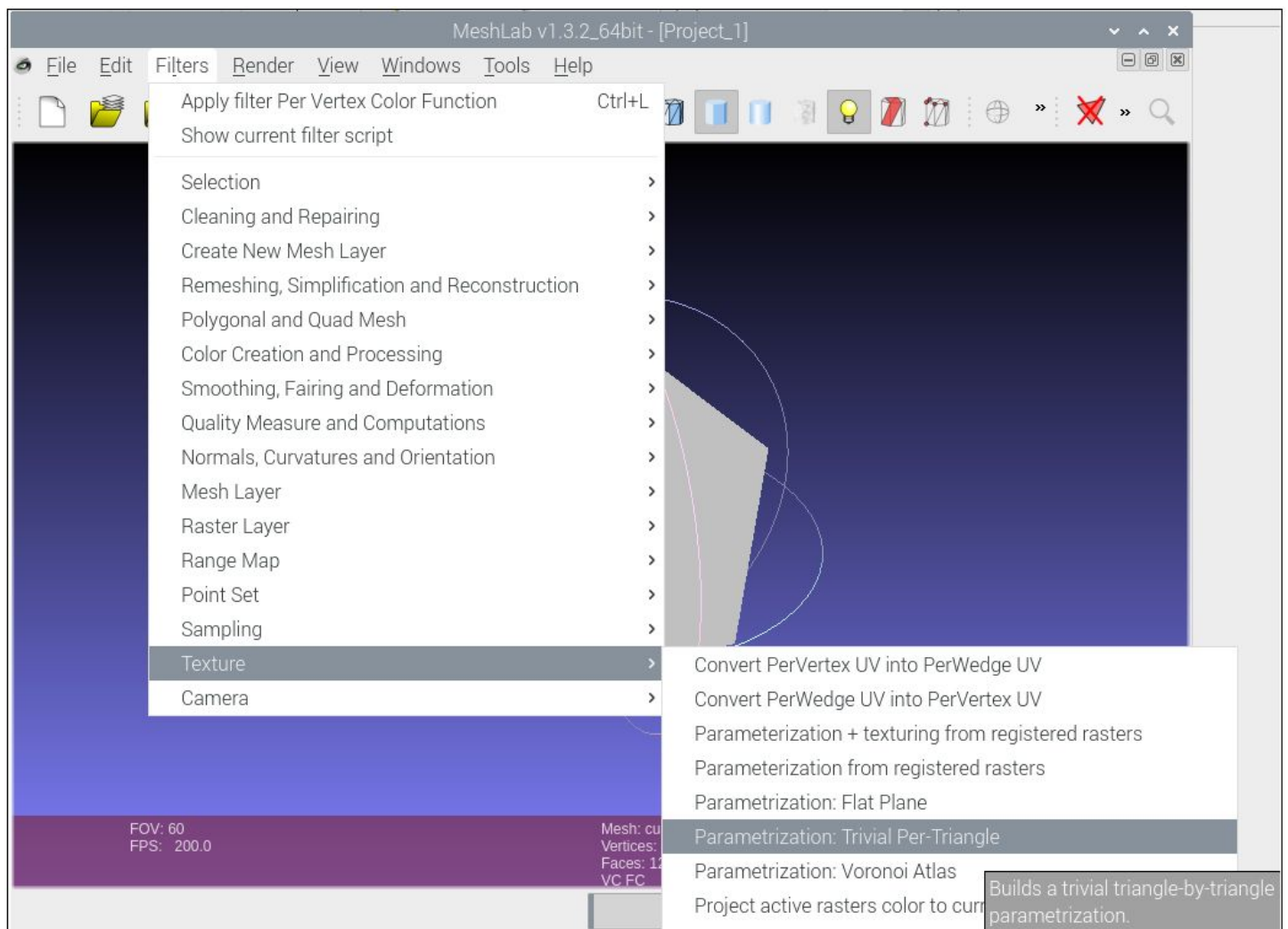


2.2- Attribuer une texture au cube dans meshlab

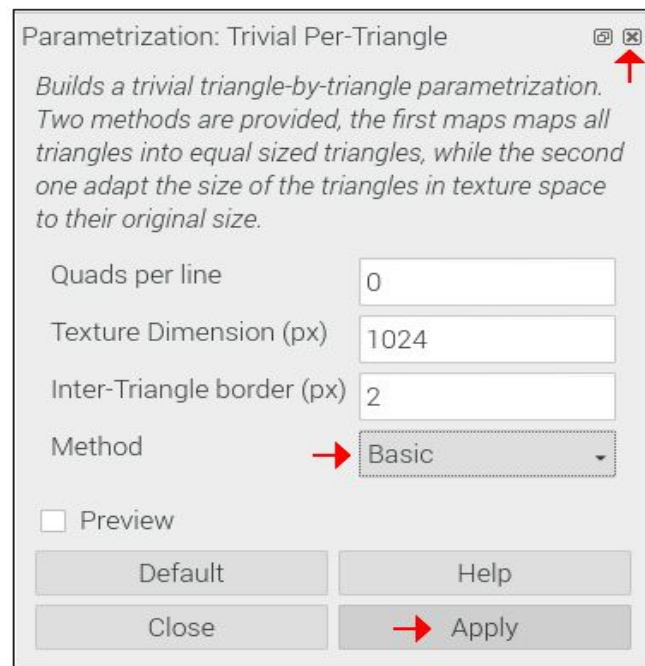
➔ On clique **Filters/Texture/Parametrization Trivial Per-triangle**



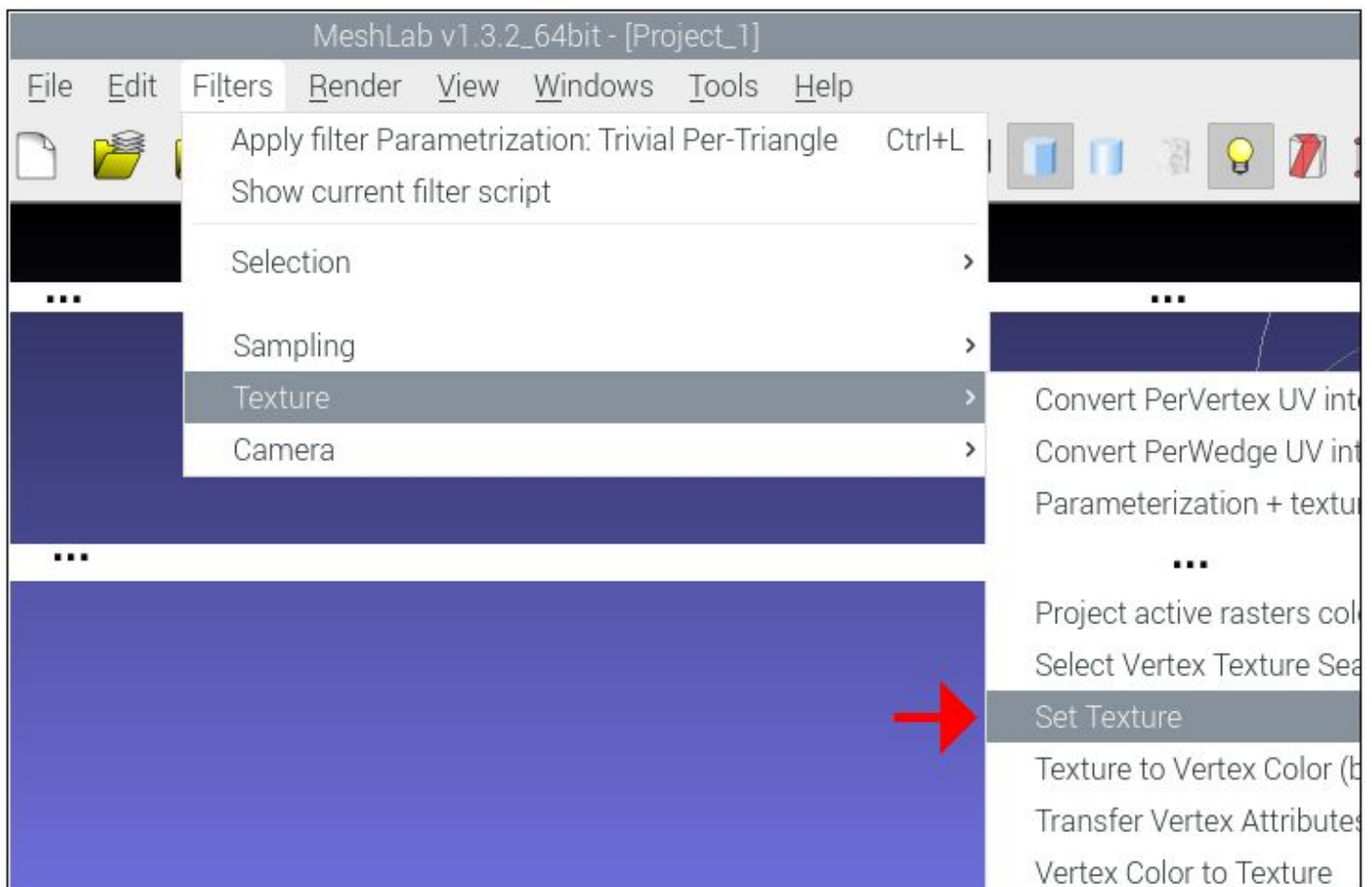
Le schéma suivant donne une vue d'ensemble de la fenêtre meshlab



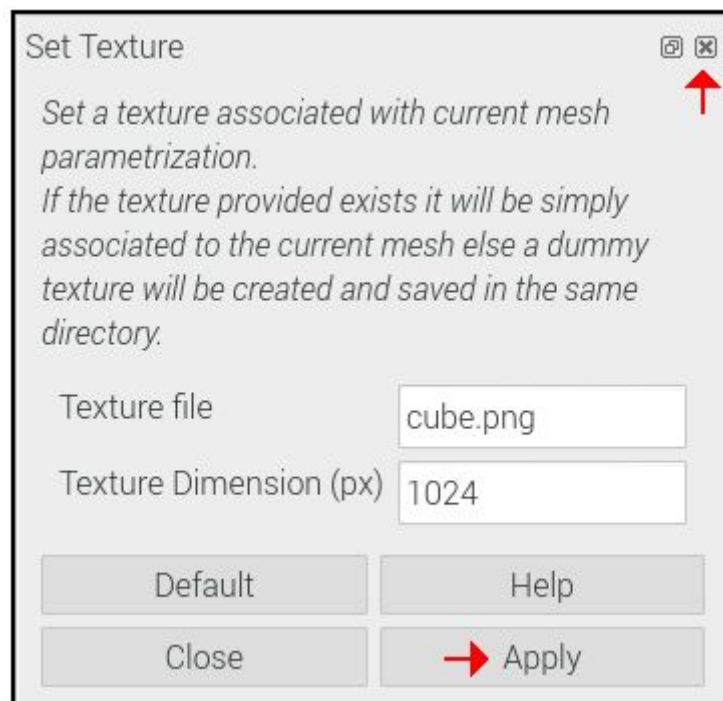
➡ Dans la boîte de dialogue qui s'ouvre, on choisit **Basic** puis on clique le bouton **Apply** puis on referme la boîte de dialogue en cliquant le bouton de fermeture contenant une croix, situé en haut à droite



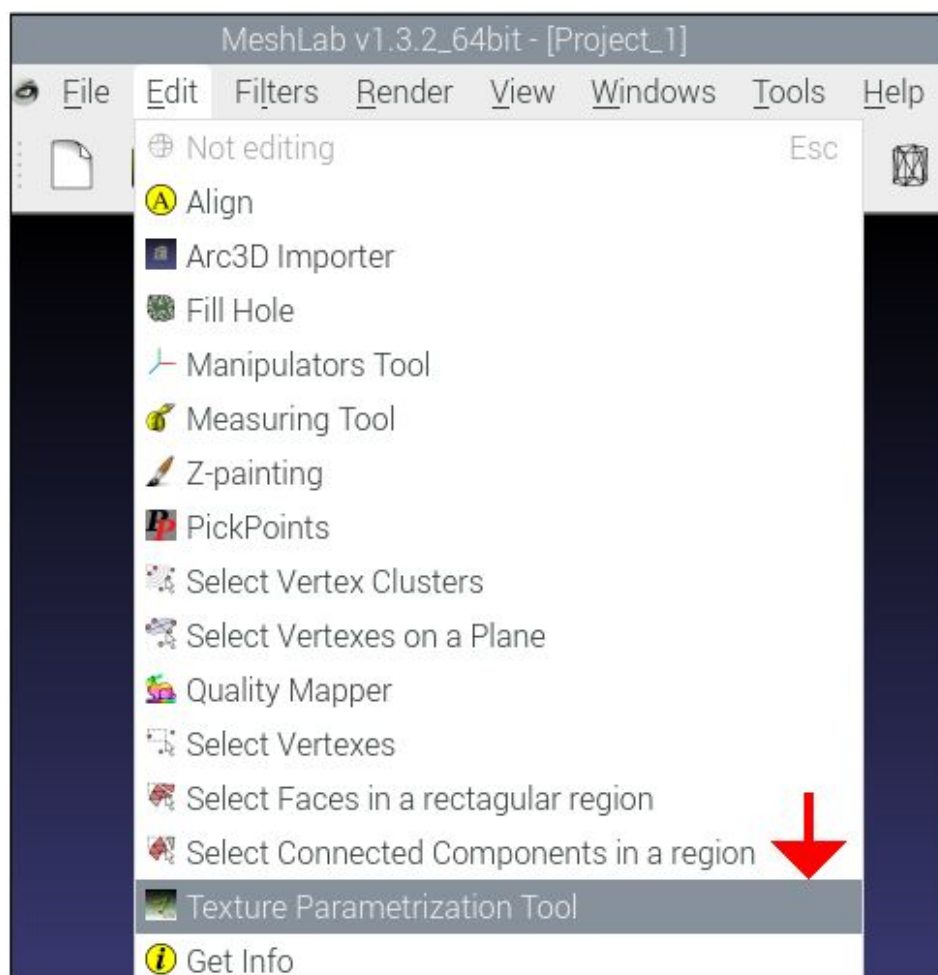
➡ On clique **Filters/Texture/Set Texture**



➡ Dans la boîte de dialogue qui s'ouvre, on clique le bouton **Apply** puis on referme la boîte de dialogue en cliquant le bouton de fermeture contenant une croix, situé en haut à droite.



➡ On clique **Edit/Texture Parametrization Tool**



➡ Dans la boîte de dialogue qui s'ouvre, on clique le bouton **Apply** puis on referme la boîte de dialogue en cliquant le bouton de fermeture contenant une croix, situé en haut à droite.

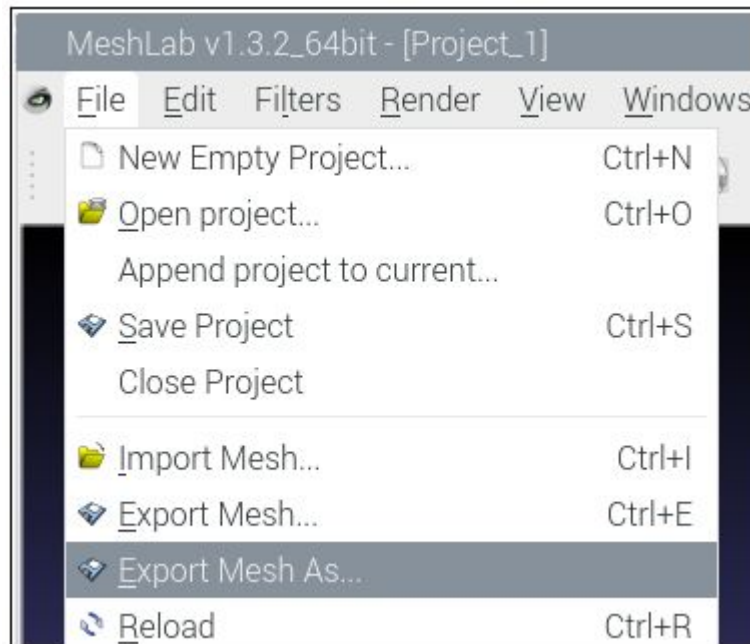


Résumé des opérations

- ➡ 1) Filters/Texture/Parametrization Trivial Per-triangle
- ➡ 2) Filters/Texture/Set Texture
- ➡ 3) Edit/Texture Parametrization Tool

2.3- Exporter le cube texturé au format obj

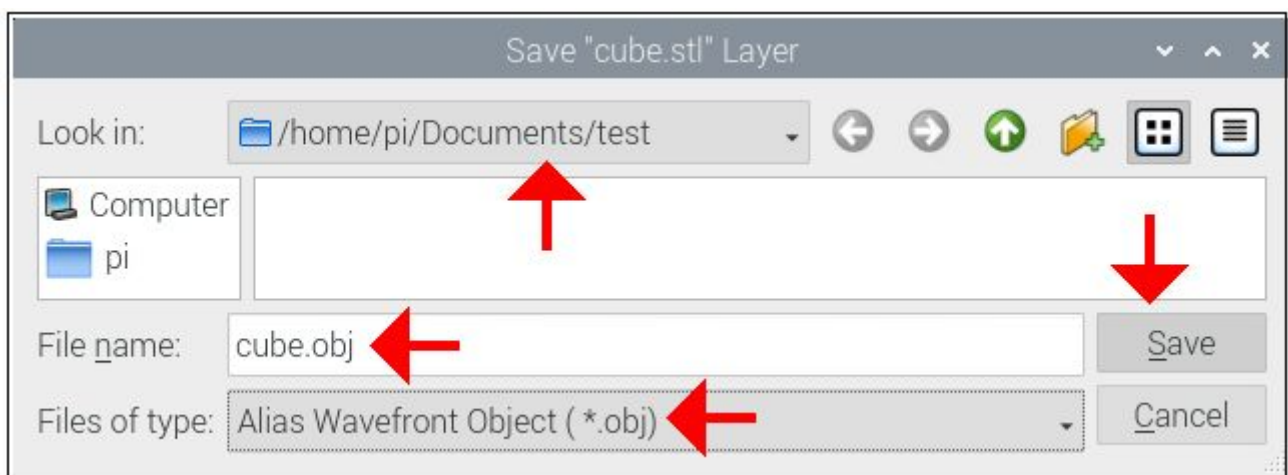
➔ On clique **File/ Export Mesh As...**



➔ Dans la boîte de dialogue qui s'ouvre, on sélectionne

- le format Wavefront Object (.obj) ;
- le nom du fichier (ici cube.obj) ;
- le répertoire de destination (ici, home/pi/Documents/test)

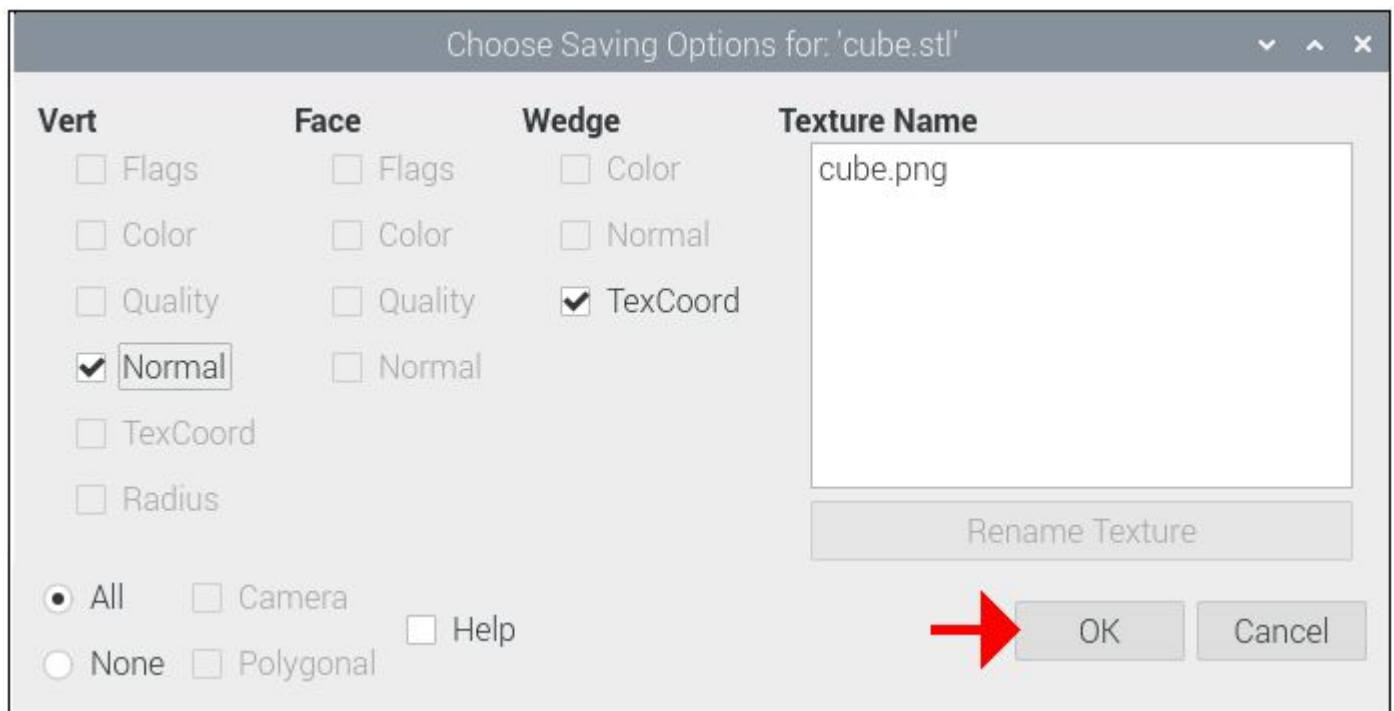
➔ puis on clique le bouton **Save**



Une boîte de dialogue s'affiche alors automatiquement, indiquant que la création de la texture est bien prise en compte.

Les indications **Wedge** (coin) **TexCoord** et **Texture Name** montrent que **meshlab** a bien pris en compte la création d'une texture associée à l'objet 3D exporté au format obj.

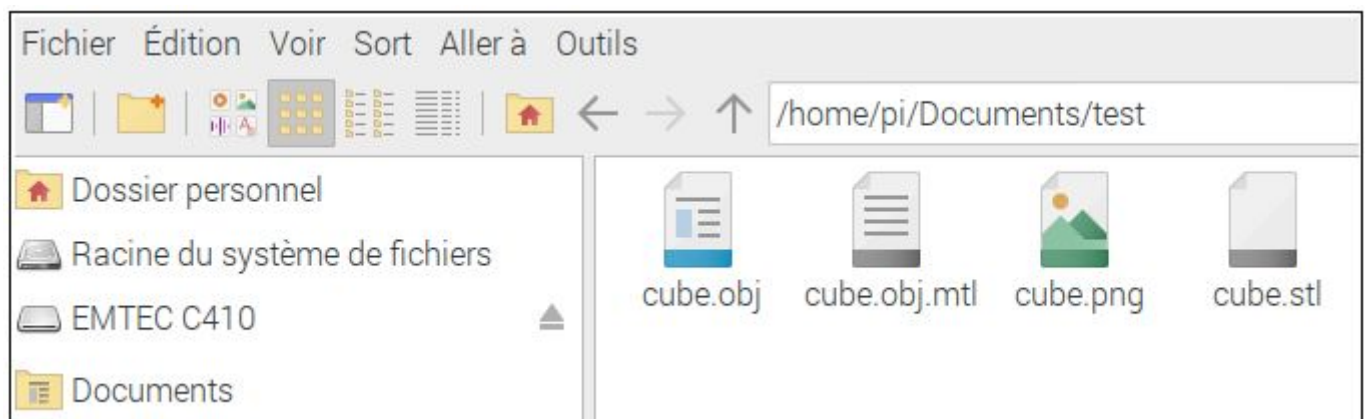
➡ On clique le bouton **OK**



➡ On se rend dans le répertoire **home/pi/Documents/test**, à l'aide de l'explorateur de fichier.

On y trouve le fichier **cube.stl** d'origine (créé avec openscad) ainsi que les trois fichiers suivants créés par **meshlab** :

- **cube.obj**: fichier contenant la description de la géométrie du cube ;
- **cube.obj.mtl**: fichier contenant la description des matériaux du cube ;
- **cube.png** : fichier contenant l'image de la texture appliquée au cube.



cube.obj

```
####  
#  
# OBJ File Generated by Meshlab  
#  
####  
# Object cube.obj  
#  
# Vertices: 8  
# Faces: 12  
#  
####  
mtllib ./cube.obj.mtl  
  
vn 0.000000 0.000000 0.785398  
v 0.000000 10.000000 10.000000  
vn 0.000000 -1.570796 0.000000  
v 0.000000 0.000000 10.000000  
vn 0.785398 0.000000 0.000000  
v 10.000000 0.000000 10.000000  
vn 1.570796 0.000000 0.000000  
v 10.000000 10.000000 10.000000  
vn 0.785398 0.000000 0.000000  
v 10.000000 10.000000 0.000000  
vn 1.570796 0.000000 0.000000  
v 10.000000 0.000000 0.000000  
vn -1.570796 0.000000 0.000000  
v 0.000000 10.000000 0.000000  
vn -0.785398 0.000000 0.000000  
v 0.000000 0.000000 0.000000  
# 8 vertices, 0 vertices normals
```

```
usemtl material_0  
vt 0.165976 0.999023  
vt 0.000977 0.834024  
vt 0.165976 0.834024  
f 1/1/1 3/2/3 4/3/4  
vt 0.167357 0.667643  
vt 0.332357 0.832643  
vt 0.167357 0.832643  
f 3/4/3 1/5/1 2/6/2  
vt 0.499309 0.999023  
vt 0.334310 0.834024  
vt 0.499309 0.834024  
f 8/7/8 5/8/5 6/9/6  
vt 0.500691 0.667643  
vt 0.665690 0.832643  
vt 0.500691 0.832643  
f 5/10/5 8/11/8 7/12/7  
vt 0.832643 0.999023  
vt 0.667643 0.834024  
vt 0.832643 0.834024  
f 8/13/8 3/14/3 2/15/2  
vt 0.834024 0.667643  
vt 0.999023 0.832643  
vt 0.834024 0.832643  
f 3/16/3 8/17/8 6/18/6  
vt 0.165976 0.665690  
vt 0.000977 0.500691  
vt 0.165976 0.500691  
f 3/19/3 5/20/5 4/21/4
```

```

vt 0.167357 0.334310
vt 0.332357 0.499309
vt 0.167357 0.499309
f 5/22/5 3/23/3 6/24/6
vt 0.499309 0.665690
vt 0.334310 0.500691
vt 0.499309 0.500691
f 5/25/5 1/26/1 4/27/4
vt 0.500691 0.334310
vt 0.665690 0.499309
vt 0.500691 0.499309
f 1/28/1 5/29/5 7/30/7
vt 0.832643 0.665690
vt 0.667643 0.500691
vt 0.832643 0.500691
f 8/31/8 1/32/1 7/33/7
vt 0.834024 0.334310
vt 0.999023 0.499309
vt 0.834024 0.499309
f 1/34/1 8/35/8 2/36/2
# 12 faces, 36 coords texture
# End of File

```

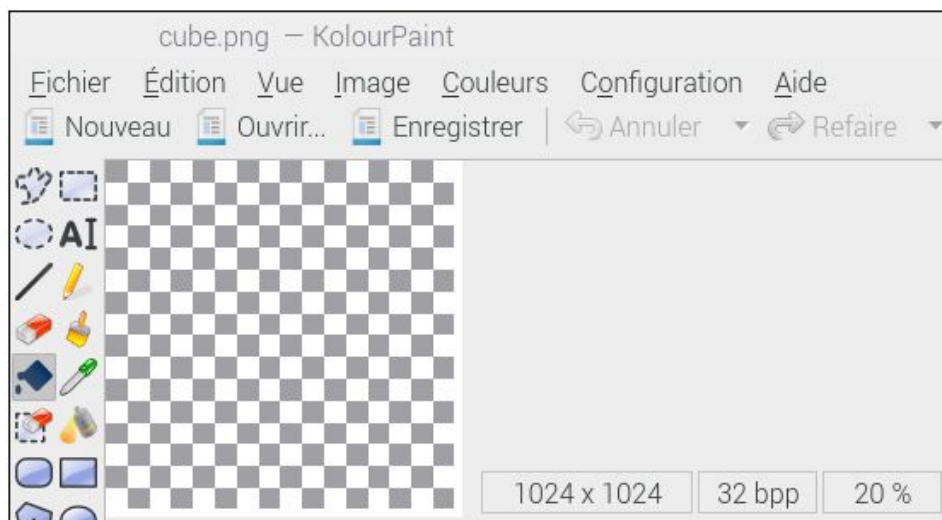
cube.obj.mtl

```

#
# Wavefront material file
# Converted by Meshlab Group
#
newmtl material_0
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube.png

```

cube.png

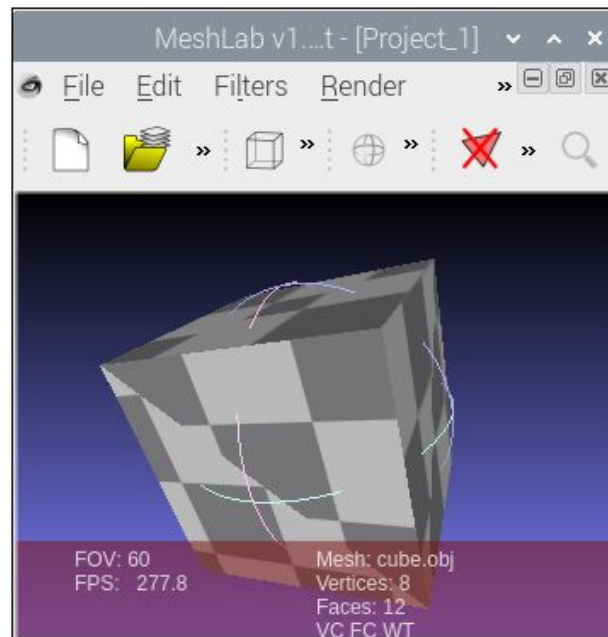


Le fichier cube.png est un fichier maquette (dummy file) créé automatiquement par meshlab. Il peut être modifié, à l'aide de **Gimp** ou de **kolour-paint** par exemple, pour contenir une image de texture de notre choix.

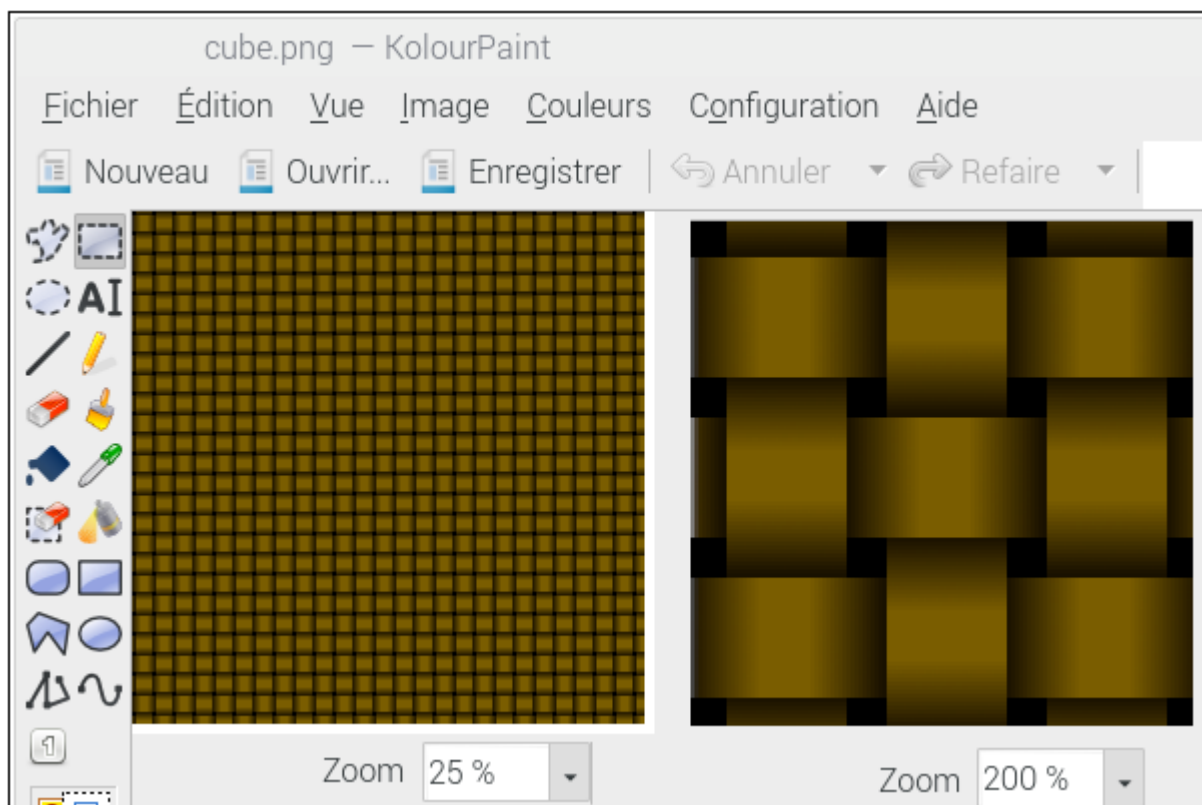
2.4 Ouvrir le fichier cube.obj avec meshlab

➔ On ouvre **meshlab** et on clique **File/Import/ cube.obj**

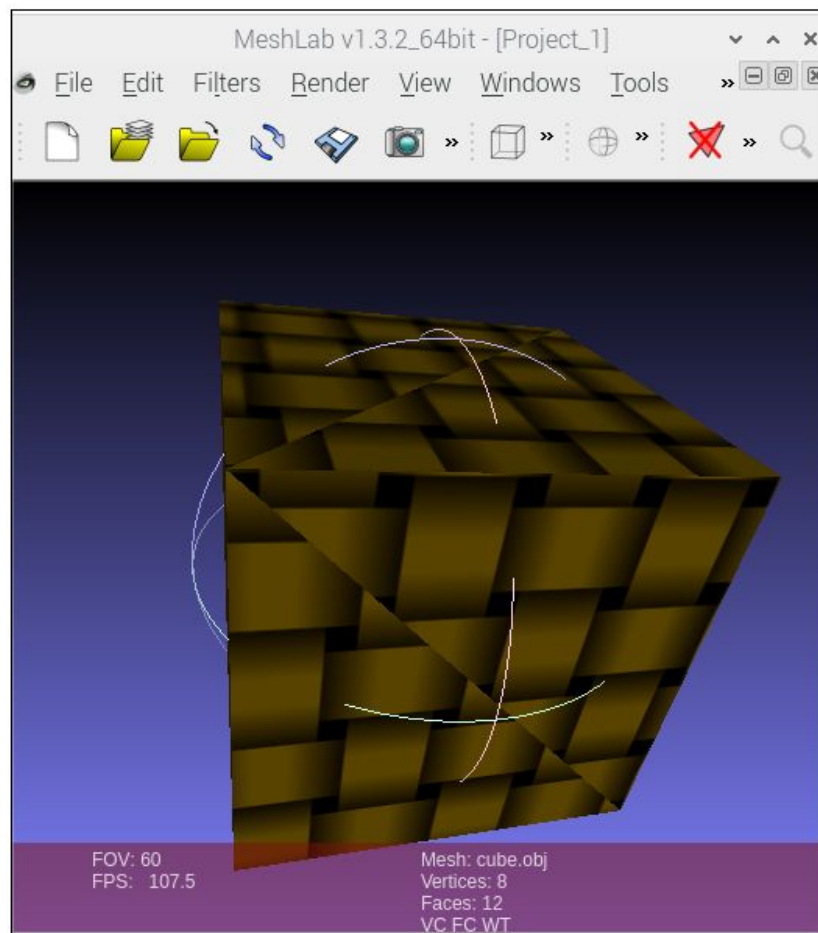
Le cube s'affiche alors automatiquement dans la fenêtre de **meshlab** recouvert de la texture **cube.png** qui avait été créée par défaut.



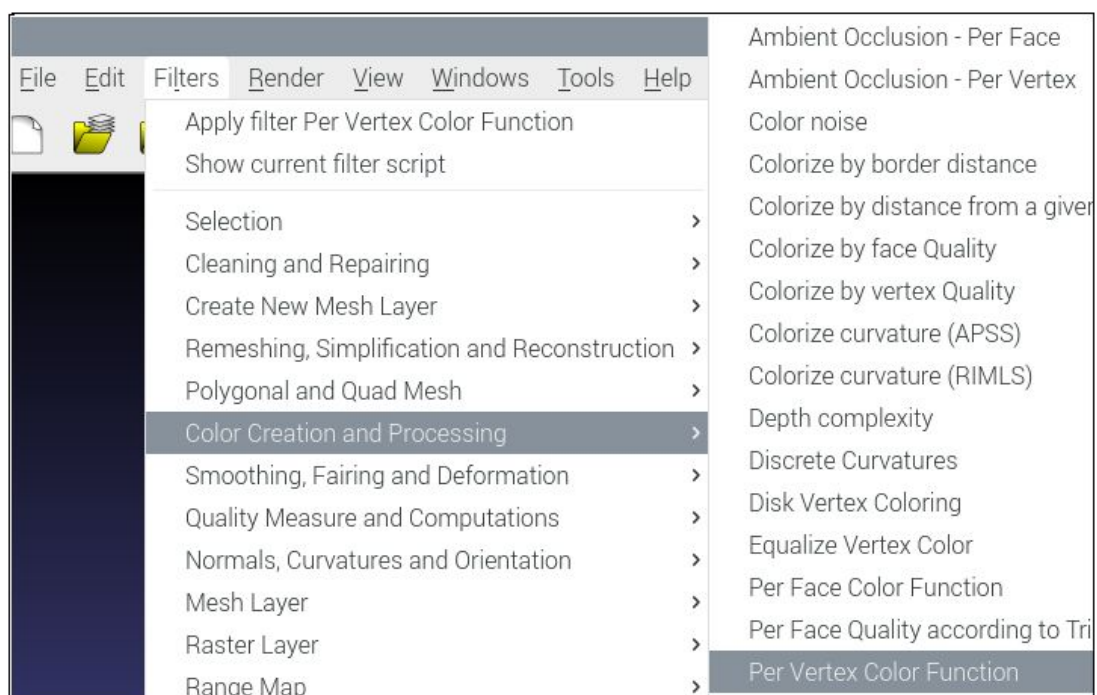
On peut modifier cette texture, à l'aide de **Gimp** ou de **Kolourpaint** par exemple. On peut par exemple créer une image en forme de tresse marron que l'on enregistre dans le fichier cube.png.



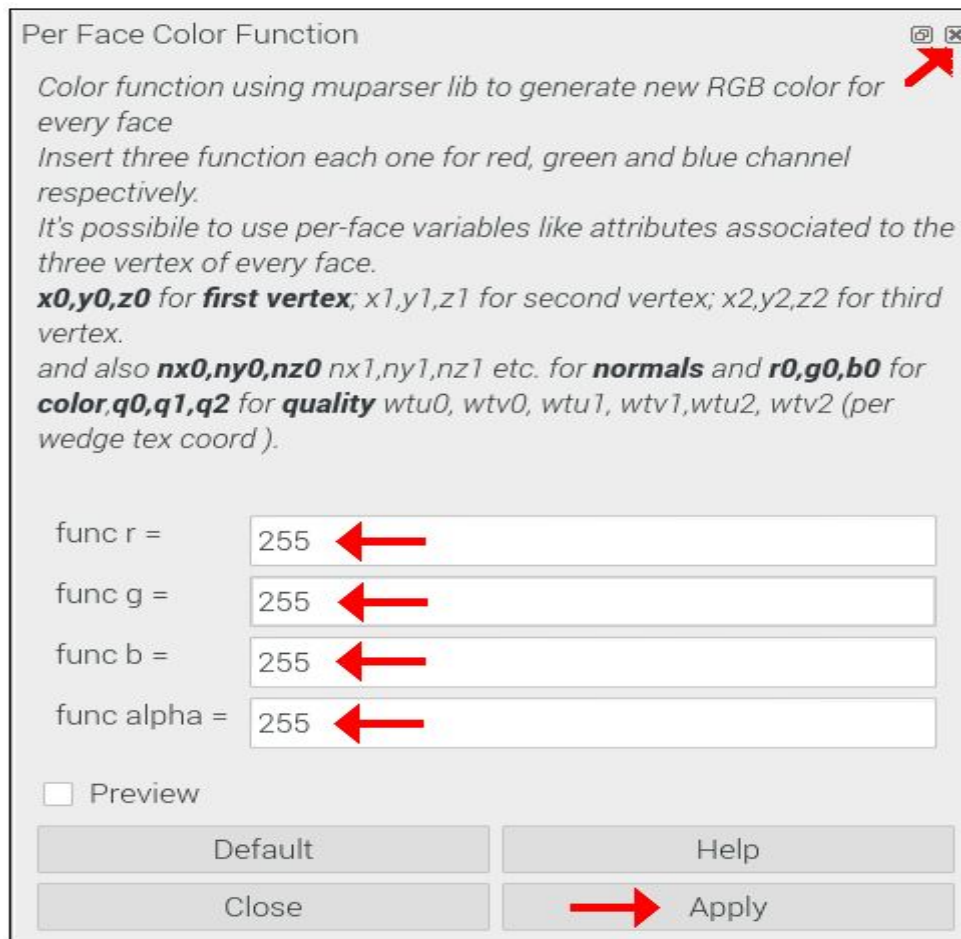
➡ On ouvre alors à nouveau le fichier **cube.obj** avec **meshlab** et on vérifie que le cube s'affiche désormais avec la texture modifiée.



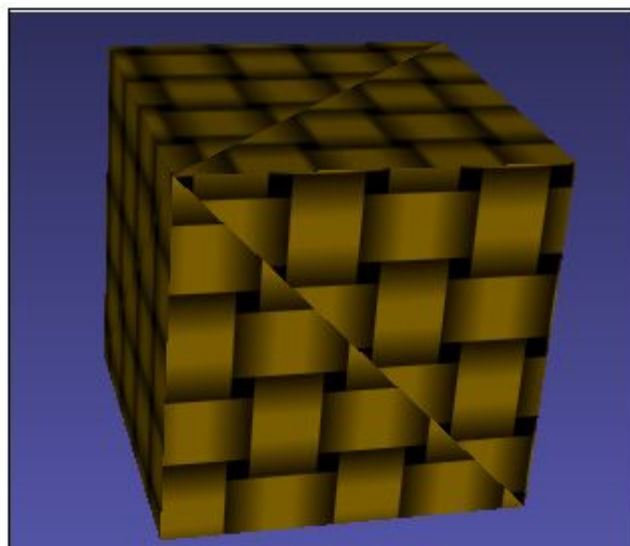
➡ Pour s'assurer que la texture du cube s'affiche correctement, on clique **Filters/ColorCreationandProcessing/ PerVertexColor Function**



Dans la boîte de dialogue qui s'ouvre, on place les valeurs 255 dans les zones **r**, **g**, **b** et **a** puis on clique le bouton **Apply** puis on referme la boîte de dialogue en cliquant le bouton de fermeture contenant une croix, situé en haut à droite.



On décoche la case **View/Trackball/ Show Trackball** afin de ne pas afficher la boule de commande de rotation (trackball) autour du cube. **Remarque** : le mappage de texture effectué automatiquement par meshlab peut être modifié ou ajusté (cf paragraphe 4)

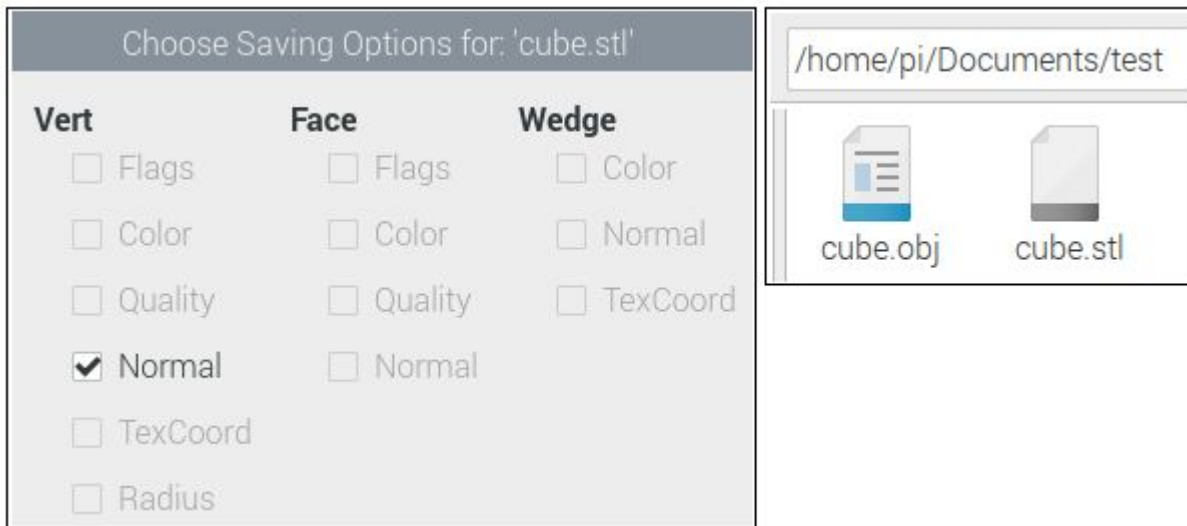


3-Tester des filtrages de cube.stl avec meshlab

3.1) Sans filtrage

- File/Import Mesh.../cube.stl
- File/Export Mesh As.../cube.obj

On voit que dans le cadre d'une importation de **cube.stl** suivie aussitôt d'une exportation dans le format **obj** (sans filtrage préalable), **meshlab** crée des informations uniquement sur les **sommets** du cube (Vertices) et non pas sur les **faces** ni sur les **coins** (Wedge). Dans ce cas, meshlab crée uniquement un fichier **cube.obj** et ne crée pas de fichier **cube.obj.mtl** ni de fichier **cube.png**.



Dans le fichier cube.obj, on trouve la description de :

- 8 sommets (v),
- 8 normales de sommets (vn)
- 12 faces (f).

Les 12 faces (f) sont en fait ici les 2x6 mailles triangulaires qui constituent les 6 faces du cube

cube.obj

```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cube.obj
#
# Vertices: 8
# Faces: 12
#
####
vn 0.000000 0.000000 0.785398
v 0.000000 10.000000 10.000000
vn 0.000000 -1.570796 0.000000
```

```

v 0.000000 0.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 0.000000 10.000000
vn 1.570796 0.000000 0.000000
v 10.000000 10.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 10.000000 0.000000
vn 1.570796 0.000000 0.000000
v 10.000000 0.000000 0.000000
vn -1.570796 0.000000 0.000000
v 0.000000 10.000000 0.000000
vn -0.785398 0.000000 0.000000
v 0.000000 0.000000 0.000000
# 8 vertices, 0 vertices normals

```

```

f 1//1 3//3 4//4
f 3//3 1//1 2//2
f 8//8 5//5 6//6
f 5//5 8//8 7//7
f 8//8 3//3 2//2
f 3//3 8//8 6//6
f 3//3 5//5 4//4
f 5//5 3//3 6//6
f 5//5 1//1 4//4
f 1//1 5//5 7//7
f 8//8 1//1 7//7
f 1//1 8//8 2//2
# 12 faces, 0 coords texture

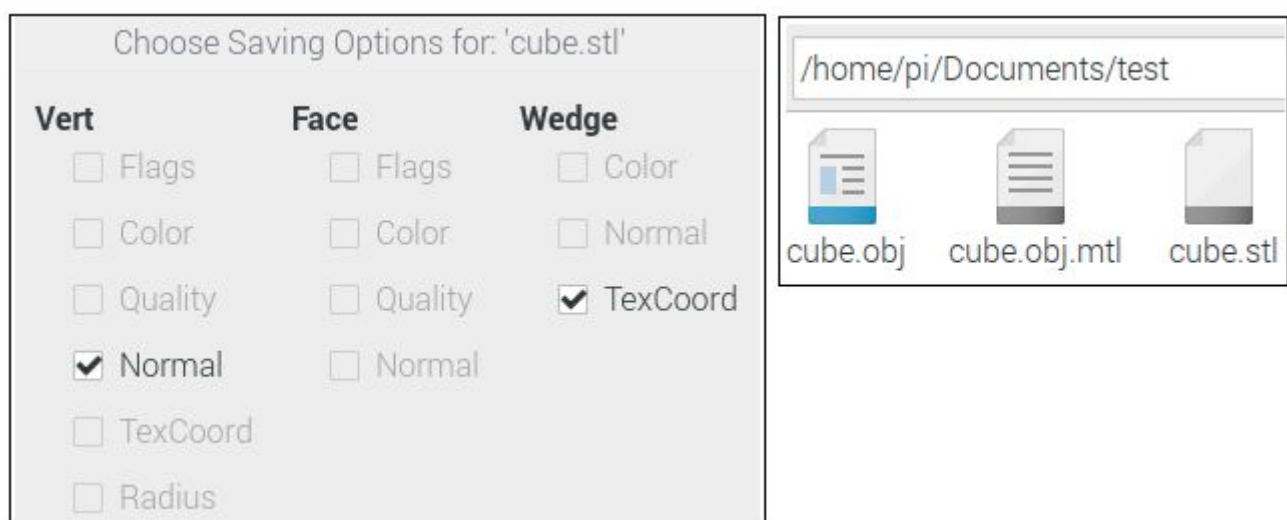
```

```
# End of File
```

3.2) Filtrage Parametrization Trivial Per-Triangle /Basic

- File/Import Mesh.../cube.stl
- Filters/Texture/Parametrization Trivial Per-Triangle /Basic
- File/Export Mesh As.../cube.obj

On voit que dans le cas d'un filtrage de type paramétrisation par triangle basique, avant l'export, **meslab** crée des coordonnées de textures.



Dans ce cas, **meshlab** crée également un fichier **cube.obj.mtl** mais cependant ne crée pas de fichier **cube.png**.

Dans le fichier **cube.obj**, on trouve un renvoi vers le fichier **cube.obj.mtl** plus la description de :

- 8 sommets (v),
- 8 normales de sommets (vn)
- 12 faces (f)
- 36 coordonnées de textures (3 coordonnées x,y,z par face)

cube.obj

```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cube.obj
#
# Vertices: 8
# Faces: 12
#
####
mtllib ./cube.obj.mtl

vn 0.000000 0.000000 0.785398
v 0.000000 10.000000 10.000000
vn 0.000000 -1.570796 0.000000
v 0.000000 0.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 0.000000 10.000000
vn 1.570796 0.000000 0.000000
v 10.000000 10.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 10.000000 0.000000
vn 1.570796 0.000000 0.000000
v 10.000000 0.000000 0.000000
vn -1.570796 0.000000 0.000000
v 0.000000 10.000000 0.000000
vn -0.785398 0.000000 0.000000
v 0.000000 0.000000 0.000000
# 8 vertices, 0 vertices normals

usemtl material_0
vt 0.000977 0.669024
vt 0.330976 0.999023
vt 0.000977 0.999023
f 1/1/1 3/2/3 4/3/4
vt 0.332357 0.997642
vt 0.002358 0.667643
vt 0.332357 0.667643
f 3/4/3 1/5/1 2/6/2
vt 0.334310 0.669024
vt 0.664309 0.999023
vt 0.334310 0.999023
f 8/7/8 5/8/5 6/9/6
vt 0.665690 0.997642
vt 0.335691 0.667643
vt 0.665690 0.667643
```

```

f 5/10/5 8/11/8 7/12/7
vt 0.667643 0.669024
vt 0.997642 0.999023
vt 0.667643 0.999023
f 8/13/8 3/14/3 2/15/2
vt 0.999023 0.997642
vt 0.669024 0.667643
vt 0.999023 0.667643
f 3/16/3 8/17/8 6/18/6
vt 0.000977 0.335691
vt 0.330976 0.665690
vt 0.000977 0.665690
f 3/19/3 5/20/5 4/21/4
vt 0.332357 0.664309
vt 0.002358 0.334310
vt 0.332357 0.334310
f 5/22/5 3/23/3 6/24/6
vt 0.334310 0.335691
vt 0.664309 0.665690
vt 0.334310 0.665690
f 5/25/5 1/26/1 4/27/4
vt 0.665690 0.664309
vt 0.335691 0.334310
vt 0.665690 0.334310
f 1/28/1 5/29/5 7/30/7
vt 0.667643 0.335691
vt 0.997642 0.665690
vt 0.667643 0.665690
f 8/31/8 1/32/1 7/33/7
vt 0.999023 0.664309
vt 0.669024 0.334310
vt 0.999023 0.334310
f 1/34/1 8/35/8 2/36/2
# 12 faces, 36 coords texture
# End of File

```

Le fichier **cube.obj.mtl** décrit le matériel (lumière, couleurs, texture) utilisé. Dans le cas présent, il ne contient que des informations de base, correspondant à un éclairage et à une couleur standards du cube.

cube.obj.mtl

```

#
# Wavefront material file
# Converted by Meshlab Group
#

newmtl material_0
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000

```

Remarque

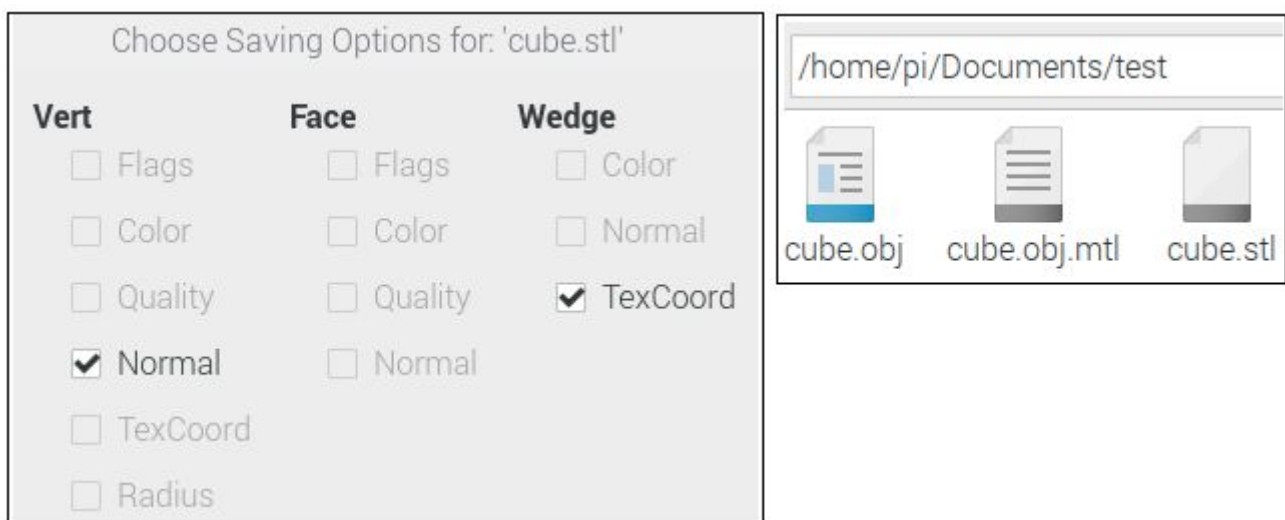
Dans notre exemple, si on effectue le traitement **Parametrization Trivial Per-Triangle /Space optimizing** au lieu du traitement **Parametrization Trivial Per-Triangle /Basic**, on obtient le même contenu pour les fichiers

cube.obj et **cube.obj.mtl**. Cela n'est pas étonnant car le cube est un objet 3D basique

3.3) Filtrage Parametrization : Flat Plane

- File/Import Mesh.../cube.stl
- Filters/Texture/Parametrization : Flat Plane
- File/Export Mesh As.../cube.obj

On voit que dans le cas d'une paramétrisation de type **Flat Plane**, avant l'export, **meslab** crée des coordonnées de textures.



Dans ce cas, meslab crée également un fichier **cube.obj.mtl** sans créer de fichier **cube.png**

Dans le fichier **cube.obj**, on trouve toujours un renvoi vers le fichier **cube.obj.mtl** mais cette fois-ci on a la description de :

- 8 sommets (v),
- 8 normales de sommets (vn)
- 12 faces (f) et, cette fois-ci, à la différence de la fois précédente,
- 4 coordonnées de textures seulement au lieu de 36.

cube.obj

```
####  
#  
# OBJ File Generated by Meshlab  
#  
####  
# Object cube.obj  
#  
# Vertices: 8  
# Faces: 12  
#  
####  
mtllib ./cube.obj.mtl
```

```

vn 0.000000 0.000000 0.785398
v 0.000000 10.000000 10.000000
vn 0.000000 -1.570796 0.000000
v 0.000000 0.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 0.000000 10.000000
vn 1.570796 0.000000 0.000000
v 10.000000 10.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 10.000000 0.000000
vn 1.570796 0.000000 0.000000
v 10.000000 0.000000 0.000000
vn -1.570796 0.000000 0.000000
v 0.000000 10.000000 0.000000
vn -0.785398 0.000000 0.000000
v 0.000000 0.000000 0.000000
# 8 vertices, 0 vertices normals

```

```

usemtl material_0
vt 0.000000 1.000000
vt 1.000000 0.000000
vt 1.000000 1.000000
f 1/1/1 3/2/3 4/3/4
vt 0.000000 0.000000
f 3/2/3 1/1/1 2/4/2
f 8/4/8 5/3/5 6/2/6
f 5/3/5 8/4/8 7/1/7
f 8/4/8 3/2/3 2/4/2
f 3/2/3 8/4/8 6/2/6
f 3/2/3 5/3/5 4/3/4
f 5/3/5 3/2/3 6/2/6
f 5/3/5 1/1/1 4/3/4
f 1/1/1 5/3/5 7/1/7
f 8/4/8 1/1/1 7/1/7
f 1/1/1 8/4/8 2/4/2
# 12 faces, 4 coords texture
# End of File

```

Le contenu du fichier cube.obj.mtl reste inchangé

cube.obj.mtl

```

#
# Wavefront material file
# Converted by Meshlab Group
#

newmtl material_0
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000

```

4- Texturer cube.obj à la main

4.1 Texturer une seule face

En partant des fichiers cube.obj et cube.obj.mtl précédents, créés par meshlab, on peut texturer l'objet 3D de façon manuelle en procédant de la façon suivante.

➡ On reprend le fichier **cube.obj** précédent dans lequel on modifie la partie `usemtl material_0` comme indiqué ci-après

cube.obj

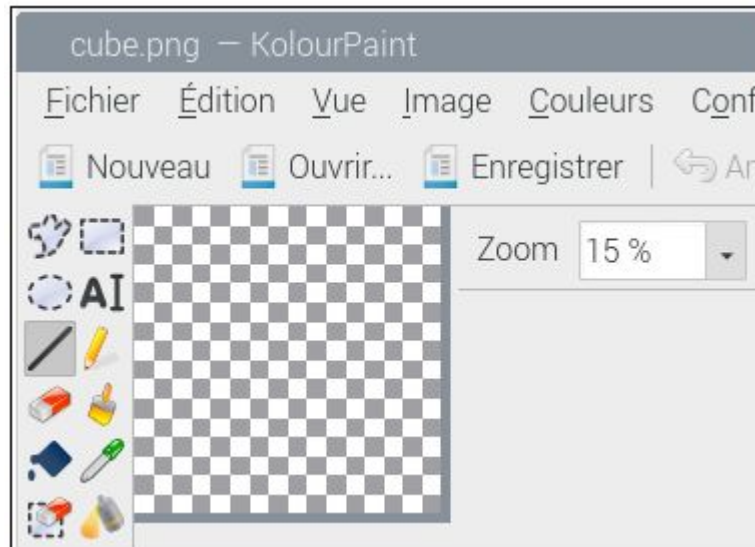
```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cube.obj
#
# Vertices: 8
# Faces: 12
#
####
mtllib ./cube.obj.mtl

vn 0.000000 0.000000 0.785398
v 0.000000 10.000000 10.000000
vn 0.000000 -1.570796 0.000000
v 0.000000 0.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 0.000000 10.000000
vn 1.570796 0.000000 0.000000
v 10.000000 10.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 10.000000 0.000000
vn 1.570796 0.000000 0.000000
v 10.000000 0.000000 0.000000
vn -1.570796 0.000000 0.000000
v 0.000000 10.000000 0.000000
vn -0.785398 0.000000 0.000000
v 0.000000 0.000000 0.000000
# 8 vertices, 0 vertices normals

usemtl material_0
vt 0.000000 1.000000
vt 0.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 1.000000
f 1/1/1 3/3/3 4/4/4
f 3/3/3 1/1/1 2/2/2
f 8//8 5//5 6//6
f 5//5 8//8 7//7
f 8//8 3//3 2//2
f 3//3 8//8 6//6
f 3//3 5//5 4//4
f 5//5 3//3 6//6
f 5//5 1//1 4//4
f 1//1 5//5 7//7
f 8//8 1//1 7//7
```

```
f 1//1 8//8 2//2
# 12 faces, 4 coords texture
# End of File
```

➡ On crée, à l’aide gimp ou de kolourpaint, un fichier image **cube.png** destiné à servir de texture pour le cube. Par exemple le fichier cube.png de taille 1024x1024 pixels suivant:

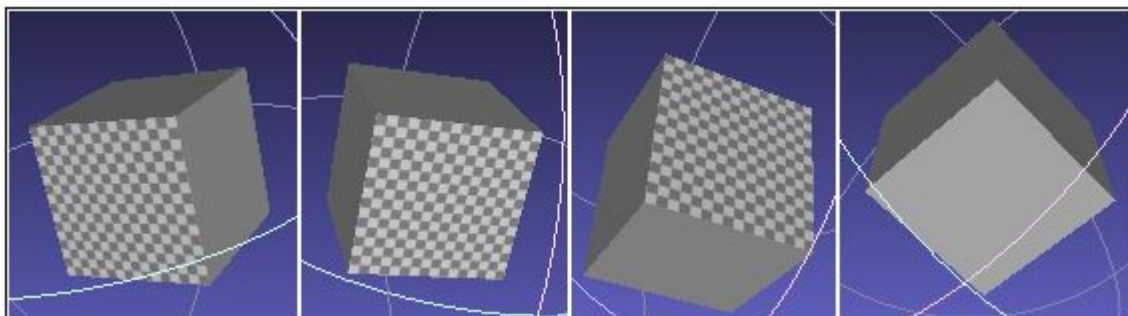


➡ On ajoute l’instruction `map_Kd cube.png` à la fin du fichier **cube.obj.mtl**

cube.obj.mtl

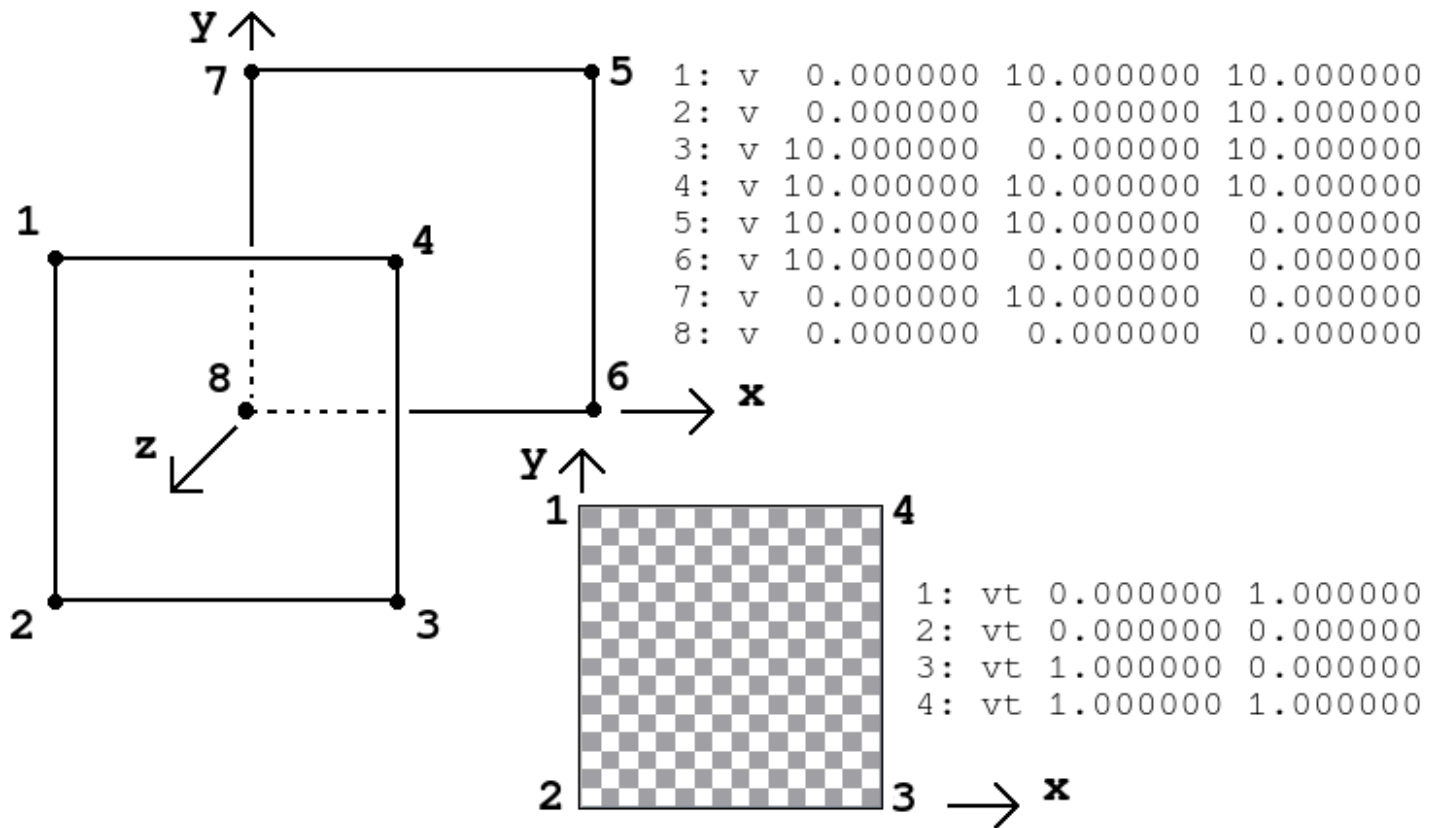
```
#
# Wavefront material file
# Converted by Meshlab Group
#
newmtl material_0
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube.png
```

En ouvrant cube.obj avec **meshlab** on constate qu’avec le mappage de texture réalisé, la face avant du cube est texturée avec l’image **cube.png**. Les autres faces (celles indiquées **v//vn**, au lieu de **v/vt/vn**) ne sont pas texturées.



4.2 Texturer les 6 faces du cube à l'identique

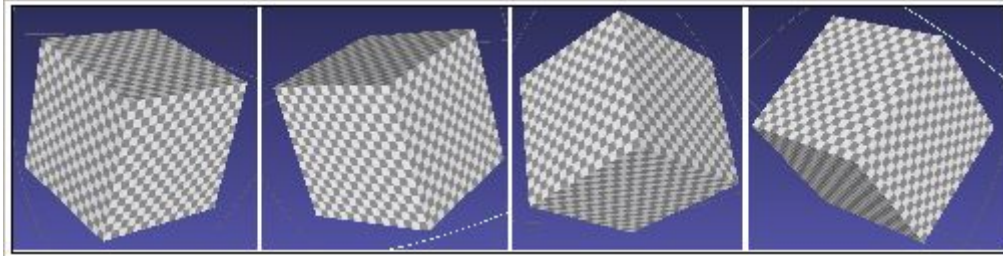
Pour manipuler la partie contenant les coordonnées de texture, on peut utiliser les schémas suivants qui illustrent l'ordre des sommets (1 à 8) et des coordonnées de textures (1 à 4) définis par **meshlab** dans le fichier **cube.obj**.



D'où le codage des faces suivant, en se plaçant en face de la face considérée

```
#face avant
f 1/1/1 3/3/3 4/4/4
f 3/3/3 1/1/1 2/2/2
#face arrière
f 8/3/8 5/1/5 6/2/6
f 5/1/5 8/3/8 7/4/7
#face du bas
f 8/4/8 3/2/3 2/3/2
f 3/2/3 8/4/8 6/1/6
#face de droite
f 3/2/3 5/4/5 4/1/4
f 5/4/5 3/2/3 6/3/6
#face de dessus
f 5/4/5 1/2/1 4/3/4
f 1/2/1 5/4/5 7/1/7
#face de gauche
f 8/2/8 1/4/1 7/1/7
f 1/4/1 8/2/8 2/3/2
```

En ouvrant cube.obj avec meshlab, on constate que chaque face du cube est texturée avec l'image contenue dans le fichier cube.png



Cube.obj

```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cube.obj
#
# Vertices: 8
# Faces: 12
#
####
mtllib ./cube.obj.mtl

vn 0.000000 0.000000 0.785398
v 0.000000 10.000000 10.000000
vn 0.000000 -1.570796 0.000000
v 0.000000 0.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 0.000000 10.000000
vn 1.570796 0.000000 0.000000
v 10.000000 10.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 10.000000 0.000000
vn 1.570796 0.000000 0.000000
v 10.000000 0.000000 0.000000
vn -1.570796 0.000000 0.000000
v 0.000000 10.000000 0.000000
vn -0.785398 0.000000 0.000000
v 0.000000 0.000000 0.000000
# 8 vertices, 0 vertices normals

usemtl material_0
vt 0.000000 1.000000
vt 0.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 1.000000
#face avant
f 1/1/1 3/3/3 4/4/4
f 3/3/3 1/1/1 2/2/2
#face arrière
f 8/3/8 5/1/5 6/2/6
f 5/1/5 8/3/8 7/4/7
#face du bas
f 8/4/8 3/2/3 2/3/2
f 3/2/3 8/4/8 6/1/6
#face de droite
f 3/2/3 5/4/5 4/1/4
f 5/4/5 3/2/3 6/3/6
```

```
#face de dessus
f 5/4/5 1/2/1 4/3/4
f 1/2/1 5/4/5 7/1/7
#face de gauche
f 8/2/8 1/4/1 7/1/7
f 1/4/1 8/2/8 2/3/2
# 12 faces, 4 coords texture
# End of File
```

cube.obj.mtl

```
#
# Wavefront material file
# Converted by Meshlab Group
#

newmtl material_0
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube.png
```

4.3 Texturer les 6 faces du cube avec 6 textures

On commence par créer 6 textures qu'on appelle cube01.png à cube06.png par exemple (une texture différente pour chaque face du cube).



On modifie le fichier cube.obj.mtl de façon à prendre en compte ces 6 textures

cube.obj.mtl

```
#
# Wavefront material file
# Converted by Meshlab Group
#

newmtl material_1
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube01.png

newmtl material_2
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
```

```

Ns 0.000000
map_Kd cube02.png

newmtl material_3
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube03.png

newmtl material_4
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube04.png

newmtl material_5
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube05.png

newmtl material_6
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube06.png

```

Il suffit alors d'insérer l'instruction `usemtl material_i` (avec `i` allant de 1 à 6) avant chaque face, dans le fichier précédent, pour obtenir le résultat souhaité.

```

####
#
# OBJ File Generated by Meshlab
#
####
# Object cube.obj
#
# Vertices: 8
# Faces: 12
#
####
mtllib ./cube.obj.mtl

vn 0.000000 0.000000 0.785398
v 0.000000 10.000000 10.000000
vn 0.000000 -1.570796 0.000000
v 0.000000 0.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 0.000000 10.000000

```

```

vn 1.570796 0.000000 0.000000
v 10.000000 10.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 10.000000 0.000000
vn 1.570796 0.000000 0.000000
v 10.000000 0.000000 0.000000
vn -1.570796 0.000000 0.000000
v 0.000000 10.000000 0.000000
vn -0.785398 0.000000 0.000000
v 0.000000 0.000000 0.000000
# 8 vertices, 0 vertices normals

vt 0.000000 1.000000
vt 0.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 1.000000

#face avant
usemtl material_1
f 1/1/1 3/3/3 4/4/4
f 3/3/3 1/1/1 2/2/2

#face arrière
usemtl material_2
f 8/3/8 5/1/5 6/2/6
f 5/1/5 8/3/8 7/4/7

#face du bas
usemtl material_3
f 8/4/8 3/2/3 2/3/2
f 3/2/3 8/4/8 6/1/6

#face de droite
usemtl material_4
f 3/2/3 5/4/5 4/1/4
f 5/4/5 3/2/3 6/3/6

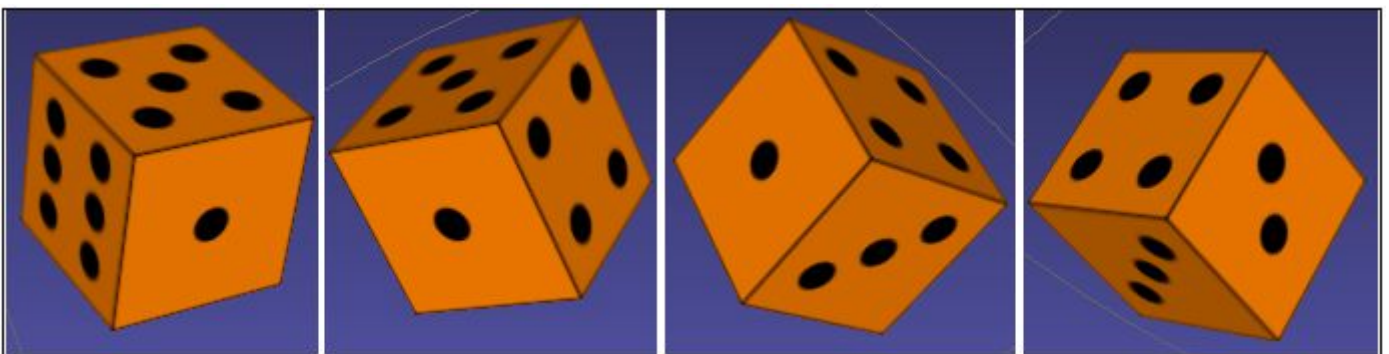
#face de dessus
usemtl material_5
f 5/4/5 1/2/1 4/3/4
f 1/2/1 5/4/5 7/1/7

#face de gauche
usemtl material_6
f 8/2/8 1/4/1 7/1/7
f 1/4/1 8/2/8 2/3/2
# 12 faces, 4 coords texture

# End of File

```

On ouvre **cube.obj** avec **meshlab** et on vérifie que le résultat obtenu est correct.

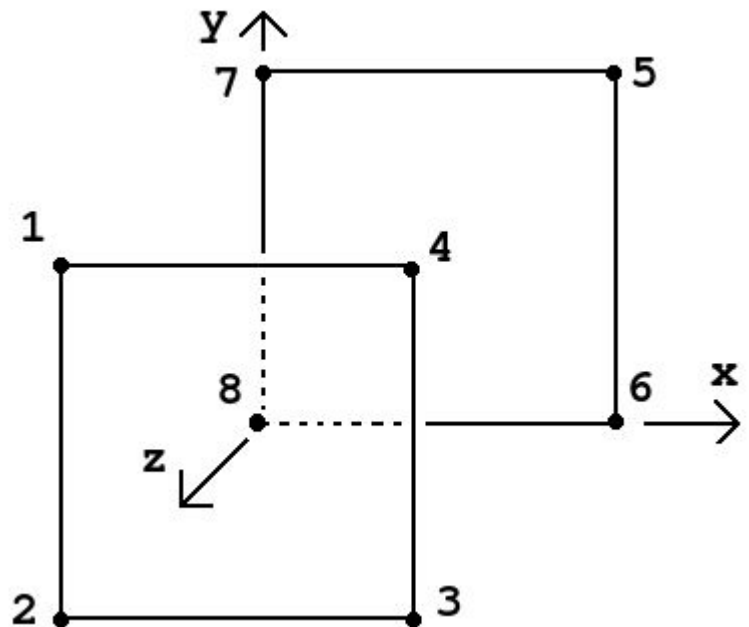
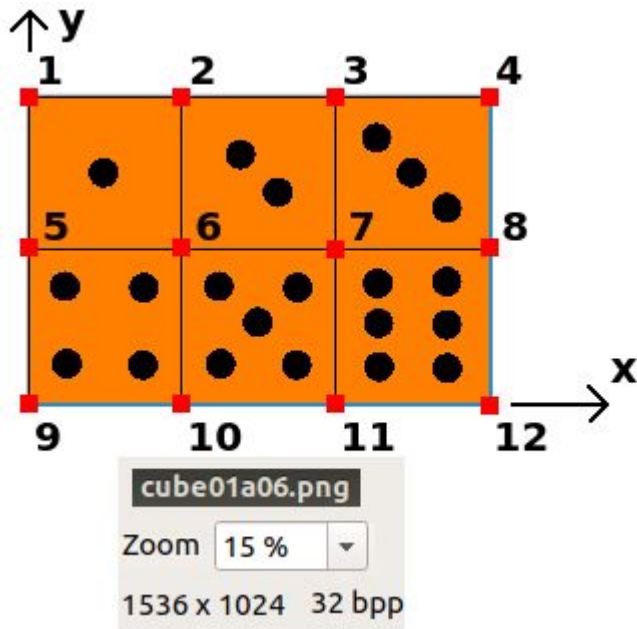


4.4 Texturer les 6 faces avec une texture globale

On crée une texture globale, à l'aide de Gimp ou de kolourpaint par exemple, qu'on enregistre dans le fichier cube01a06.png.

Si on dimensionne la texture de chaque face du cube à 512x512 pixels, la texture globale fait 1536x1024 pixels.

Dans ce cas, on a les 12 couples **vt** (x,y) suivants de coordonnées de texture. On en déduit les faces **v/vt/vn** du cube avec vt et vn restant inchangés par rapport au fichier obj initial généré par meshlab.



```
01: vt 0.000000 1.000000
02: vt 0.333333 1.000000
03: vt 0.666666 1.000000
04: vt 1.000000 1.000000
05: vt 0.000000 0.500000
06: vt 0.333333 0.500000
07: vt 0.666666 0.500000
08: vt 1.000000 0.500000
09: vt 0.000000 0.000000
10: vt 0.333333 0.000000
11: vt 0.666666 0.000000
12: vt 1.000000 0.000000
```

```
#face avant [1]
f 1/1/1 3/6/3 4/2/4
f 3/6/3 1/1/1 2/5/2

#face arrière [2]
f 8/7/8 5/2/5 6/6/6
f 5/2/5 8/7/8 7/3/7

#face du bas [3]
f 8/7/8 3/4/3 2/3/2
f 3/4/3 8/7/8 6/8/6

#face de droite [4]
f 3/9/3 5/6/5 4/5/4
f 5/6/5 3/9/3 6/10/6

#face de dessus [5]
f 5/7/5 1/10/1 4/11/4
f 1/10/1 5/7/5 7/6/7

#face de gauche [6]
f 8/11/8 1/8/1 7/7/7
f 1/8/1 8/11/8 2/12/2
# 12 faces, 12 coords texture
```

Les fichiers cube.obj et cube.obj.mtl résultant sont les suivants.

cube.obj

```
####
#
# OBJ File Generated by Meshlab
#
####
# Object cube.obj
#
# Vertices: 8
# Faces: 12
#
####
mtllib ./cube.obj.mtl

vn 0.000000 0.000000 0.785398
v 0.000000 10.000000 10.000000
vn 0.000000 -1.570796 0.000000
v 0.000000 0.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 0.000000 10.000000
vn 1.570796 0.000000 0.000000
v 10.000000 10.000000 10.000000
vn 0.785398 0.000000 0.000000
v 10.000000 10.000000 0.000000
vn 1.570796 0.000000 0.000000
v 10.000000 0.000000 0.000000
vn -1.570796 0.000000 0.000000
v 0.000000 10.000000 0.000000
vn -0.785398 0.000000 0.000000
v 0.000000 0.000000 0.000000
# 8 vertices, 0 vertices normals

usemtl material_1
vt 0.000000 1.000000
vt 0.333333 1.000000
vt 0.666666 1.000000
vt 1.000000 1.000000
vt 0.000000 0.500000
vt 0.333333 0.500000
vt 0.666666 0.500000
vt 1.000000 0.500000
vt 0.000000 0.000000
vt 0.333333 0.000000
vt 0.666666 0.000000
vt 1.000000 0.000000

#face avant [1]
f 1/1/1 3/6/3 4/2/4
f 3/6/3 1/1/1 2/5/2

#face arrière [2]
f 8/7/8 5/2/5 6/6/6
f 5/2/5 8/7/8 7/3/7

#face du bas [3]
f 8/7/8 3/4/3 2/3/2
f 3/4/3 8/7/8 6/8/6
```

```
#face de droite [4]
f 3/9/3 5/6/5 4/5/4
f 5/6/5 3/9/3 6/10/6

#face de dessus [5]
f 5/7/5 1/10/1 4/11/4
f 1/10/1 5/7/5 7/6/7

#face de gauche [6]
f 8/11/8 1/8/1 7/7/7
f 1/8/1 8/11/8 2/12/2
# 12 faces, 12 coords texture
# End of File
```

cube.obj.mtl

```
#
# Wavefront material file
# Converted by Meshlab Group
#
newmtl material_1
Ka 0.200000 0.200000 0.200000
Kd 1.000000 1.000000 1.000000
Ks 1.000000 1.000000 1.000000
Tr 1.000000
illum 2
Ns 0.000000
map_Kd cube01a06.png
```

En ouvrant cube.obj avec meshlab on obtient le même résultat qu'avec l'exemple précédent qui utilisait un fichier png différent pour chaque face du cube.

