



---

# Rapport de stage : Photogrammétrie

---

par Annie Hip-Ki  
L2 informatique  
Année 2013/2014

Encadrement

Maîtres de stage : Carole Blanc

Pascal Desbarats

du 7 juillet 2014 au 22 août 2014



# Remerciements

Tout d'abord, je tiens à remercier Pascal Desbarats pour avoir proposé plusieurs sujets de stage à ses étudiants. J'ai donc pu choisir un sujet de stage parmi une gamme de sujets intéressants concernant l'imagerie numérique.

Je remercie aussi Carole Blanc, mon maître de stage, qui a aussi sélectionné le même sujet, d'avoir choisi d'encadrer notre binôme. Elle nous a permis, en collaboration avec Pascal Desbarats, d'avancer dans le stage en nous guidant et en nous aidant à surmonter les problèmes rencontrés.

Enfin, je remercie mon partenaire de stage Timothé Barbaux pour son soutien dans le travail, la qualité de ses recherches et ses compétences en mathématiques. Il m'a beaucoup aidé lors de ce stage et travailler avec lui sur ce sujet a été une bonne expérience.

# Table des matières

<b>Introduction</b>	<b>5</b>
<b>1 Cadre du stage</b>	<b>6</b>
<b>2 Objectifs du stage</b>	<b>7</b>
<b>3 État de l’art</b>	<b>8</b>
3.1 Définition . . . . .	8
3.2 Types . . . . .	8
3.3 Domaines d’application . . . . .	9
3.4 Historique . . . . .	9
3.5 Principe . . . . .	10
3.5.1 Détection de points d’intérêt et descripteurs . . . . .	10
3.5.2 Mise en correspondance des points . . . . .	12
3.5.3 Reconstruction 3D . . . . .	13
<b>4 Outils existants</b>	<b>14</b>
4.1 Langages, bibliothèques . . . . .	14
4.1.1 Python . . . . .	14
4.1.2 OpenCV . . . . .	14
4.2 Logiciels de photogrammétrie . . . . .	14
4.2.1 Autodesk 123D . . . . .	14
4.2.2 Correlator3D . . . . .	15
4.3 Logiciels de modélisation 3D . . . . .	15
4.3.1 CATIA . . . . .	15
4.3.2 Maya . . . . .	15
4.3.3 3ds Max . . . . .	16
4.3.4 3DReshaper . . . . .	16

4.3.5	Blender . . . . .	16
4.3.6	Geomagic . . . . .	16
4.3.7	Polyworks . . . . .	17
4.3.8	SketchUp . . . . .	17
4.3.9	Wings 3D . . . . .	17
<b>5</b>	<b>Réalisation</b>	<b>18</b>
5.1	Introduction au problème . . . . .	18
5.1.1	Lecture d'un fichier et calculs . . . . .	18
5.1.2	Représentation visuelle . . . . .	19
5.2	Généralisation de l'angle de prise de vue . . . . .	21
5.3	Résultats . . . . .	22
	<b>Conclusion</b>	<b>23</b>
	<b>Bibliographie</b>	<b>24</b>

# Introduction

J'ai obtenu ce stage grâce à mon professeur d'Initiation à l'Imagerie Numérique, Pascal Desbarats, qui a proposé plusieurs sujets de stages concernant, par exemple, l'imprimante 3D, le scanner 3D ou la photogrammétrie. Il était aussi possible de proposer soi-même son sujet de stage. La photogrammétrie m'a interpellée bien que je ne savais pas exactement ce que c'était. Je savais que le travail sur les images 3D et 2D allait me plaire et ce sujet collait parfaitement à mon projet professionnel concernant entre autres les images numériques. Ce stage de recherche s'est déroulé sur deux mois, du 7 juillet au 22 août 2014.

Je commencerai par présenter le cadre du stage puis définirai les objectifs du stage. Je présenterai ensuite la photogrammétrie et les outils existants pour finir par la partie d'application des objectifs.

La partie "État de l'art" a été faite en collaboration avec Timothé Barbaux. La partie "Outils existants" a été faite à deux et est donc identique sur les deux rapports. La partie réalisation a été faite ensemble puis séparée en deux parties complémentaires.

# 1 Cadre du stage

Le LaBRI (Laboratoire Bordelais de Recherche en Informatique), anciennement "Laboratoire de Mathématiques et d'Informatique de Bordeaux", a été créé en 1988 (cf. Figure 1.1). Il est actuellement dirigé par Pascal Weil.



FIGURE 1.1 – Le LaBRI en 2014

Il rassemble plus de 300 personnes issues de différents partenariats tels que l'Université de Bordeaux ou le CNRS (Centre National de la Recherche Scientifique) . On y trouve des enseignants chercheurs, des chercheurs, du personnel administratif et technique ainsi que des doctorants, post-doctorants ou ingénieurs contractuels.

Le laboratoire regroupe six équipes de recherches telles que l'équipe Image et Son au sein de laquelle j'ai effectué mon stage. <sup>1</sup>

---

1. [http ://www.labri.fr/](http://www.labri.fr/)

## 2 Objectifs du stage

Pendant les premières semaines, l'objectif du stage a été de dresser un état de l'art sur le sujet du stage : la photogrammétrie. La recherche s'est faite en binôme. Il fallait chercher et comprendre les méthodes pour reconstituer, à partir de plusieurs images 2D, une image en 3D. Ensuite il a fallu rechercher les outils nécessaires à la réalisation de ces méthodes.

À la suite de ces recherches, nous sommes partis d'un objet 3D duquel nous devons créer des images 2D correspondant à des photographies qui auraient pu être prises dans des plans orthogonaux à cet objet. Puis les tâches ont été divisées et devaient permettre d'aboutir au produit final, l'image en 3D. Il fallait à partir de plusieurs photographies d'un personnage, prises à des angles déterminés, reconstruire le personnage en objet 3D. Cependant, à cause de certains problèmes rencontrés, nous sommes restés sur la partie consistant à créer des images 2D.



## 3 État de l'art

Dans cette section, je présenterai le concept de la photogrammétrie et détaillerai en particulier la partie sur laquelle je travaillerai dans la seconde partie du stage.

### 3.1 Définition

La photogrammétrie vient des mots grecs "photos" (lumière), "gramma" (signe, écrit) et "métron" (mesure). [3] Elle permet de reconstituer une image 3D à partir d'images 2D représentant le même sujet sous différents angles. Elle se base sur le phénomène de reconstitution du relief par les yeux. Chaque oeil perçoit une image 2D d'un angle différent et le cerveau utilise la technique de stéréoscopie afin de retrouver toutes les profondeurs des points réels, c'est-à-dire à quelle distance se situent les points par rapport à nos yeux.

### 3.2 Types

Il existe 3 types de photogrammétrie.

La **photogrammétrie analogique** utilise des photographies sur film argentique et la reconstruction 3D se fait principalement par des méthodes optiques et mécaniques.

La **photogrammétrie analytique** utilise le même type d'images que l'analogique mais repose sur une reconstruction 3D mathématique via des ordinateurs.

La **photogrammétrie numérique** est le traitement le plus avantageux. Elle utilise des images numériques et donne un résultat sous forme d'un fichier numérique. [1] [5]

### 3.3 Domaines d'application

On utilise essentiellement la photogrammétrie en architecture, en topographie, en archéologie ou encore en médecine, en robotique ou dans les enquêtes policières. [4]

### 3.4 Historique

Le français Aimé Laussedat, considéré comme l'inventeur de la photogrammétrie, entreprenait des recherches sur la perspective en 1849 en se basant sur une idée de Charles-François Beautemps-Beaupré. Il développa ainsi "la méthode des perspectives".[3]

On doit le mot "photogrammétrie", inventé en 1893, à l'allemand Albrecht Meydenbauer qui travaillait sur la géométrie descriptive. [3]

## 3.5 Principe

Pour reconstituer une image en 3D, il faut passer par plusieurs étapes. Tout d'abord, il faut chercher les points clés d'une image nécessaires à la reconstruction. Par exemple, pour obtenir un visage, on aura besoin de la forme des yeux, l'emplacement du nez, de la bouche... Ensuite, il faut retrouver ces mêmes points clés sur les autres images mises à disposition qui ont été prises sous un autre angle. Une fois les points détectés, il faudra alors les mettre en correspondance, savoir où se situe un même point sur deux images différentes. C'est sur cette partie que mes recherches se sont basées. La seconde partie, sur laquelle mon collègue s'est centré est la reconstruction 3D. Pour des informations plus détaillées, voir son rapport de stage. [2]

### 3.5.1 Détection de points d'intérêt et descripteurs

Les points d'intérêt d'une image sont en général les coins d'un objet ou les contours marquant un changement de texture, par exemple la séparation d'un arbre et du ciel.

La méthode la plus connue pour détecter les points d'intérêt est celle de Harris. [11] Quelques temps après, Lowe, chercheur canadien, a amélioré l'idée d'Harris. La méthode de Lowe permet d'avoir une plus grande précision car elle détecte en plus des points d'intérêt, les régions d'intérêt. [10] L'algorithme qu'a conçu Lowe s'intitule Scale-invariant feature transform (SIFT). Cet algorithme permet aussi la mise en correspondance des points contrairement à celui de Harris, c'est-à-dire que sa méthode sait trouver un même point dans deux images différentes représentant le même objet. [12]

Dans la première étape de la méthode SIFT, on cherche d'abord à trouver les points clés. Ensuite, pour pouvoir les apparier, on stocke différentes informations telles que la luminosité ou l'angle d'observation. Ces informations doivent être indépendantes de l'échelle de l'objet. On les appelle les descripteurs SIFT.

Dans un premier temps, on traite l'image en lui appliquant des filtres gaussiens pour la lisser et donc éliminer certains points qui auraient été détectés à tort en tant que points d'intérêt. On applique différentes valeurs de

seuil ( $\sigma$ ) pour les filtres gaussiens, puis on effectue une différence entre deux images consécutives afin d'obtenir une nouvelle image qui contiendra des points d'intérêt plus fiables. Ceci constitue un premier filtrage qu'on appelle «octave». Plus on filtre, plus la taille (donc résolution) diminue (cf. Figure 3.1).

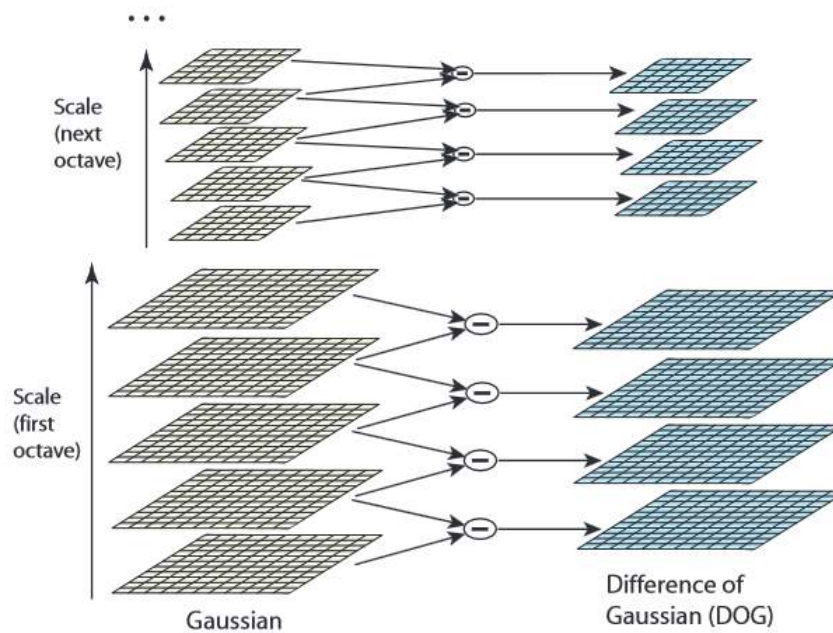


FIGURE 3.1 – Illustration de la succession de filtrages et de la différence entre deux images consécutives <sup>2</sup>.

On doit effectuer plusieurs filtrages avant d'avoir les points d'intérêt recherchés (les points invariants à l'échelle, l'angle de prise de vue, la luminosité). [6]

Dans un second temps, on applique un algorithme qui marque tous les points clés. Cependant, certains doivent être éliminés car ils pourraient fausser la reconstruction 3D, n'étant pas assez «stables». On doit éliminer les points les moins contrastés et ceux situés sur une arête.

2. <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

Dans un troisième temps, on doit chercher l'orientation des points clés gardés à l'étape précédente. Pour ce faire, on se base sur la direction des gradients dans le voisinage d'un point. On doit aussi trouver le facteur d'échelle.

Enfin, pour calculer un descripteur, il faut construire l'histogramme correspondant au point en utilisant les orientations des points voisins. [12] Pour cela, on choisit un point P et on divise son voisinage en 16 blocs de 4x4 pixels. Pour construire l'histogramme constitué de 4x4 blocs, on calcule le gradient, soit le changement de niveau de gris de chaque point clé. [9] (cf. Figure 3.2)

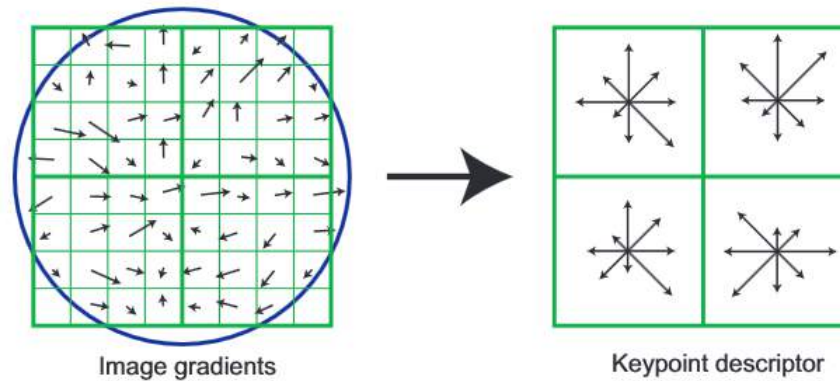


FIGURE 3.2 – Descripteur d'un point clé sur un échantillon de 8x8 pixels. Les flèches sur la partie gauche de l'image représentent les gradients de l'échantillon.<sup>3</sup>

### 3.5.2 Mise en correspondance des points

La mise en correspondance des points est la dernière étape de l'algorithme SIFT.

On cherche des points communs entre les descripteurs des points. Lorsqu'il y a au moins 3 correspondances, les points peuvent être considérés comme les mêmes. [12]

3. <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

Il faut d'abord chercher le meilleur candidat d'un point sur l'image d'origine sur la deuxième image. En général, c'est le point le plus proche en terme de descripteurs parmi tous les points clés trouvés sur la deuxième image. On utilise pour cela des algorithmes de recherche exhaustive. [7]

### **3.5.3 Reconstruction 3D**

#### **Détermination des coordonnées**

À partir des points appariés sur les différentes images 2D, nous devons trouver les points réels correspondant à des points sur l'objet 3D. Pour cela il faudra résoudre des équations mathématiques.

On détermine un point comme point d'origine ce qui nous donnera les coordonnées 2D de trois points dans chacune des deux images dont les points sont appariés. Ceci permet alors de créer des plans image qui serviront à faire de la trigonométrie (on travaille sur des photos prises orthogonalement à l'objet à des angles précis). On retrouvera alors l'équation qui nous fera trouver les 3 points correspondants dans un repère 3D ainsi que tous les autres points appariés.

#### **Modélisation**

À partir d'une liste de coordonnées 3D, on peut utiliser un logiciel de modélisation qui nous montrera le rendu en 3D. Par la suite, il sera possible de texturer et lisser l'objet 3D. De nombreux logiciels permettent de faire cela comme présentés dans la prochaine section.

## 4 Outils existants

### 4.1 Langages, bibliothèques

#### 4.1.1 Python

Créé en 1990 par Guido van Rossum, Python, langage de programmation orientée objet, est idéal pour travailler sur les images 2D. C'est celui que nous utiliserons pendant le stage. Python offre un large choix de bibliothèques permettant l'accès à de nombreux algorithmes.<sup>4</sup>

#### 4.1.2 OpenCV

OpenCV (Open Computer Vision) est une bibliothèque contenant un grand nombre de fonctions permettant de faire du traitement d'images fixes ou animées. Elle existe pour les langages C++, Python et Java. Elle contient des fonctions permettant d'appliquer la méthode SIFT ou même celle de SURF mais ces deux dernières ne sont pas libres et ne sont pas disponibles dans l'environnement de travail dont nous disposons.<sup>5</sup>

### 4.2 Logiciels de photogrammétrie

#### 4.2.1 Autodesk 123D

Autodesk a créé une gamme de logiciels téléchargeables via leur site web permettant de manipuler la 2D et la 3D.

---

4. <https://python.org/>

5. <http://opencv.org/>

123D Catch est un logiciel de photogrammétrie permettant à partir de plusieurs photographies prises sous différents angles, de reproduire l'objet en 3D.

Il existe également d'autres logiciels spécialisés comme 123D Design qui permet de créer des modèles 3D ou 123D Make qui permet à partir d'un modèle 3D, d'imprimer des pièces à assembler qui représenteront ce modèle.<sup>6</sup>

### 4.2.2 Correlator3D

Correlator3D est un logiciel développé par SimActive spécialisé dans la reconstruction 3D à partir d'images aériennes, satellites ou prises par des drones. Ce logiciel n'est pas disponible pour tout le monde. Il faut demander une démo via le site web de SimActive.<sup>7</sup>

## 4.3 Logiciels de modélisation 3D

Il existe de nombreux logiciels avec lesquels on peut reconstruire un objet 3D. Nous avons sélectionné en priorité les plus célèbres et ceux qui permettent de reconstruire un objet à partir d'un nuage de points. Nous présenterons brièvement plusieurs d'entre eux.

### 4.3.1 CATIA

CATIA, Conception Assistée Tridimensionnelle Interactive Appliquée, est un logiciel développé par Dassault Systèmes principalement utilisé dans le design industriel. Nombreuses sont les entreprises d'aéronautique utilisant CATIA. C'est un des logiciels les plus utilisés et il a permis de développer une vaste gamme de produits partout dans le monde. La version étudiante du logiciel coûte environ 100 euros.<sup>8</sup>

### 4.3.2 Maya

Maya, logiciel d'animation 3D, développé par Autodesk est écrit en C++, MEL et Python. Les prix du logiciel varient selon la version entre 800 et 4000

---

6. <http://www.123dapp.com/>

7. <http://www.simactive.com/>

8. <http://www.3ds.com>



euros la licence. Il est essentiellement utilisé dans le cinéma d'animation notamment par DreamWorks ou encore pour les effets spéciaux des films comme Harry Potter ou Iron Man.<sup>9</sup>

### 4.3.3 3ds Max

3ds Max, logiciel de modélisation, animation et rendu 3D est une référence dans le monde de l'infographie. Il est essentiellement utilisé dans la création de jeux vidéo. Par exemple, Ubisoft a utilisé 3ds Max pour le jeu Assassin's Creed. Selon la version du logiciel, la licence peut aller de 4000 à 6000 euros. Une version d'évaluation gratuite est cependant valable 30 jours.<sup>10</sup>

### 4.3.4 3DReshaper

3DReshaper est un logiciel utilisant des nuages de points issus de scanners 3D ou même de la photogrammétrie. Il est disponible seulement en version d'essai pour 1 mois et nécessite la complétion d'un formulaire pour recevoir un code.<sup>11</sup>

### 4.3.5 Blender

Blender est un logiciel libre créé en 1994 et écrit principalement en langage de programmation C/C++. C'est l'un des logiciels open source les plus populaires dans le monde pour le travail en 3D. Il peut être utilisé aussi bien pour le cinéma que pour les jeux vidéos.<sup>12</sup>

### 4.3.6 Geomagic

Geomagic est une compagnie de développement logiciel centrée sur la 3D. Elle propose entre autres Geomagic Design Direct qui permet l'édition de maillages 3D et la création de modèles 3D. Ce logiciel est surtout utilisé dans la fabrication pour améliorer ou créer un objet.<sup>13</sup>

---

9. <http://www.autodesk.fr/>

10. <http://www.autodesk.fr/>

11. <http://www.3dreshaper.com/>

12. <http://www.blender.org/>

13. <http://www.geomagic.com/fr/>

### 4.3.7 Polyworks

Polyworks est une gamme de logiciels de traitement d'image détenue par Innovmetric, entreprise québécoise. Polyworks Modeler est un logiciel qui modélise à partir d'un nuage de points. Ce logiciel est payant et ne peut pas s'acheter en ligne mais dans des points de vente un peu partout dans le monde.<sup>14</sup>

### 4.3.8 SketchUp

SketchUp Pro est un logiciel que l'on peut acquérir pour un peu moins de 500 euros. Il est plus centré sur la création 3D. La version gratuite Google SketchUp offre la possibilité de créer des modèles 3D simples. Chaque outil sélectionné et documenté dans une fenêtre à côté permettant de guider nos actions.<sup>15</sup>

### 4.3.9 Wings 3D

Wings 3D est un logiciel libre écrit dans le langage de programmation Erlang en 2001. Son interface est facile à utiliser et offre de nombreux outils de modélisation.<sup>16</sup>

---

14. <http://www.innovmetric.com/fr>

15. <http://www.sketchup.com/fr>

16. <http://www.wings3d.com/>

## 5 Réalisation

### 5.1 Introduction au problème

L'implémentation de la méthode SIFT comme il était prévu de faire n'a pas pu être faite car les fonctions disponibles sur OpenCV en rapport avec SIFT ne sont en fait pas disponibles gratuitement et non installées sur l'environnement de travail sur lequel nous avons travaillé. Les encadrants de stage nous ont donc proposé d'aller plus loin dans le travail déjà commencé.

Pour compléter l'objectif du stage, nous avons pris le problème à l'envers : de la 3D vers la 2D. Nous sommes partis d'un objet 3D pour en extraire deux "photographies" à des angles différents. Nous avons choisi de travailler d'abord sur un cube car c'est une figure simple avec peu de points clés. Nous avons choisi d'extraire les photographies à 0 degré (vue de face) et celle à 45 degrés.

Par la suite, notre tâche sera divisée en deux parties. Je traiterai la représentation graphique des faces tandis que mon collègue généralisera l'angle de prise de vue.

#### 5.1.1 Lecture d'un fichier et calculs

Tout d'abord, nous avons analysé le format d'un fichier ".obj" car c'est un format de fichier très utilisé. Celui-ci contient une liste de points, une liste de textures, une liste de normale et une autre de faces.

La partie qui nous intéresse est celle des points se présentant comme dans l'exemple qui suit :

```
v 0.0 0.0 0.0  
v 0.0 0.0 1.0
```

Chaque nombre correspond à une coordonnée dans l'espace.

On doit alors dans un premier temps rechercher les lignes commençant exclusivement par v et ignorer les lignes commençant par vt (textures) vn (normales) ou f (faces).

Pour coder cela en python, nous avons ouvert le fichier .obj et lu ligne par ligne avec la fonction "readlines" et en ajoutant des conditions de sélection de ligne. Au fur et à mesure qu'une ligne était valide (commençant par "v"), on stockait les coordonnées dans des variables (x0, y0, z0) puis on calculait son image. Après avoir obtenu l'image d'un point, on convertissait les coordonnées du point en 2D et on les stockait dans une liste. Pour cela, on utilisait la fonction "append".

Voici un extrait de code représentant la transformation de coordonnées 3D en 2D pour le plan à 45 degrés :

```
coord2d = creerTableau(2)
[...]
x = x0 + 10 #valeur changeant avec l'équation du plan voulu
y = y0
z = z0 + 10
coord2d[0] = int(round(x*math.sin(math.radians(45))
    +z*math.cos(math.radians(45)),0))
#angle de 45 degrés soit -pi/4 rad
coord2d[1] = int(round(-y,0))
```

### 5.1.2 Représentation visuelle

Enfin, nous avons représenté les résultats dans une image créée grâce à la PIL (Python Image Library). Nous récupérons les coordonnées 2D stockées précédemment dans un tableau afin d'allumer les pixels de l'image. Nous avons grossi le cube 100 fois pour mieux voir le résultat puisque les coordonnées correspondaient à un cube d'une unité de côté (cf. Figure 5.1).

#### Lecture d'un fichier

Un fichier .obj contient une partie listant les faces de l'objet. Elles sont en général triangulaires. Dans ce type de fichier, les faces sont représentées par

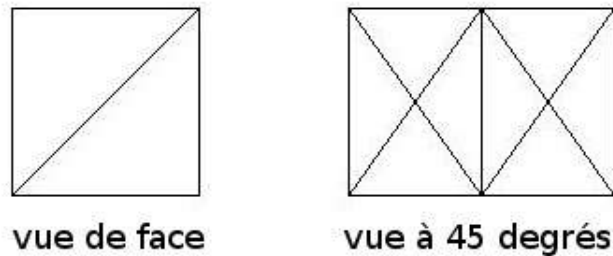


FIGURE 5.1 – Représentations 2D d'un cube

un f suivi de trois chiffres correspondant aux numéros des points constituant l'objet.

```
f 1 2 4
f 1 3 4
```

On trouve aussi des fichiers .obj sous une autre forme :

```
f 1/2/2 2/13/3 4/3/1
f 1/2/2 3/14/4 4/3/1
```

Le premier chiffre est le numéro du point, le second est la coordonnée de texture et le dernier est une normale. Ce modèle revient donc trois fois pour chaque ligne.

### Arêtes et faces

Pour représenter les arêtes, j'ai utilisé une fonction de la PIL (Python Imaging Library) avec le module ImageDraw qui permet de dessiner des lignes. Voici un extrait de code pour représenter les faces (vides) :

```
N = 3000 #pour que l'objet rentre dans l'image
im = Image.new("L", (10000,10000),"white")
draw = ImageDraw.Draw(im)
#Recherche des faces dans le fichier .obj
for ligne in lignes:
    chaine = ligne
    if chaine[0]== "f":
        if chaine[1] == " ":
```

```

#Separation des points dans le cas "f n1 n2 n3"
a= chaine.split()
i = int(a[1])-1
j = int(a[2])-1
k = int(a[3])-1
draw.line(((tableau[i][0])+N, (tableau[i][1])+N,
(tableau[j][0])+N, (tableau[j][1])+N))
draw.line(((tableau[i][0])+N, (tableau[i][1])+N,
(tableau[k][0])+N, (tableau[k][1])+N))
draw.line(((tableau[j][0])+N, (tableau[j][1])+N,
(tableau[k][0])+N, (tableau[k][1])+N))

```

Le problème dans cette partie est que toutes les arêtes sont représentées or il faudrait cacher les arêtes arrières. Par exemple, dans la vue à 45 degrés de Figure 5.1, on devrait voir uniquement deux des quatre diagonales représentées.

Pour ne pas représenter les faces cachées, il faudrait une notion de profondeur. Cependant, on ne peut pas décider de relier un point à un autre s'il sont plus en avant que deux autres points. D'après quelques recherches, il serait possible de cacher les faces par une méthode d'éclairage.[8]

L'idée de solution trouvée et non implémentée pour le moment est de calculer les normales de chaque face par un produit vectoriel (si elles ne sont pas déjà fournies dans le fichier .obj), les faces étant ordonnées dans un fichier .obj.

Puis, il faudra alors vérifier si les normales sont positives ou négatives. Si elles sont positives, elles appartiennent à l'intervalle 0 degré -180 degrés, correspondant à une image 2D, tant pour la largeur que la hauteur de l'objet et donc les faces pourront être affichées. Sinon, on ne les représente pas graphiquement.

## 5.2 Généralisation de l'angle de prise de vue

Après avoir travaillé sur des objets différents à 0 et 45 degrés, il est intéressant de voir l'objet sous un autre angle. C'est pour cela qu'il a fallu généraliser l'angle de prise de vue.

Pour connaître la démarche, voir section 4.4 du rapport de stage de Timothé Barbaux. [2]

### 5.3 Résultats

Après avoir généralisé l'angle de prise de vue, voici ce qu'on obtient pour un objet 3D plus complexe que le cube : Figure 5.2. Il est difficile de bien voir la rotation autour de l'objet car la quantité d'arêtes représentées peuvent troubler notre perception.

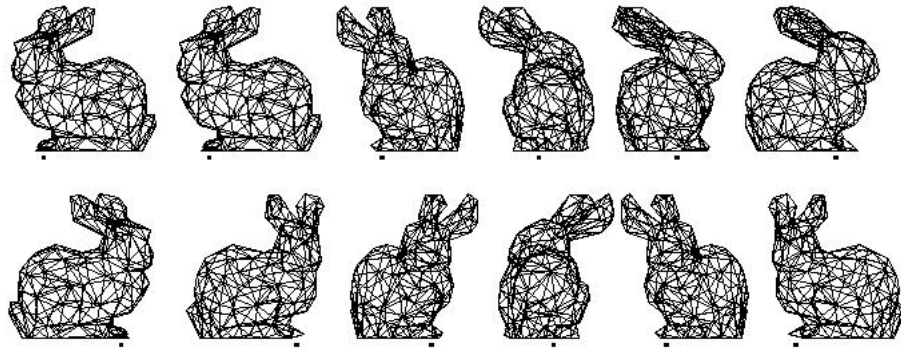


FIGURE 5.2 – Représentations 2D d'un lapin tous les 30 degrés à partir de 0.

# Conclusion

Les recherches et les résultats trouvés peuvent servir de données synthétiques pour des tests de photogrammétrie. Par exemple, grâce au petit programme créé, on peut extraire des images 2D d'un objet 3D sous l'angle choisi par l'utilisateur. Les deux rapports (celui-ci et celui de Timothé Barboux [2]) complémentaires sur ce sujet peuvent servir de base de recherche pour un éventuel autre stage sur le même sujet ou de documentation synthétique sur la photogrammétrie.

Ce stage pourrait être facilement amélioré en poussant davantage les recherches et en apportant plus de solutions aux problèmes. On pourrait essayer d'implémenter toutes les pistes possibles trouvées, chercher un algorithme de remplacement à SIFT ou encore pouvoir utiliser le programme pour des fichiers autres que celui que nous avons choisi (le fichier .obj) comme le format .stl, .3ds ou .ply.

Avoir deux encadrants est plus intéressant puisque l'on peut obtenir plusieurs avis. Les réunions hebdomadaires ont été très enrichissantes et utiles pour ce stage et pour moi-même.



# Bibliographie

- [1] photogrammétrie. <http://cmapspublic.ihmc.us/rid=1HRMCRX9F-27TZVH-YYB/Photogramm%C3%A9trie.cmap>. Consulté le 21 juillet 2014.
- [2] Timothé Barbaux. Rapport de stage : Photogrammétrie, 2014.
- [3] L'Insa de Strasbourg. Historique de la photogrammétrie architecturale. <http://www.insa-strasbourg.fr/fr/photogrammetrie>. Consulté le 15 juillet 2014.
- [4] Institut National d'Optique. Photogrammétrie. <http://www.ino.ca/fr/technologies/syst%C3%A8me-dimagerie-3d/photogramm%C3%A9trie/>. Consulté le 22 juillet 2014.
- [5] F. Gervais. Éléments de photogrammétrie : Cours de photo-interprétation, février 2010. <http://www.gecweb.info/pi/download/Polycop.pdf>. Consulté le 21 juillet 2014.
- [6] Mahzad Kalantari and Franck Jung. Estimation automatique de l'orientation relative en imagerie terrestre. *revue XYZ*, 114, 1er trimestre 2008. [http://hal.archives-ouvertes.fr/docs/00/33/47/77/PDF/XYZ\\_114\\_def.pdf](http://hal.archives-ouvertes.fr/docs/00/33/47/77/PDF/XYZ_114_def.pdf). Consulté le 21 juillet 2014.
- [7] David Lowe. Distinctive image features from scale-invariant keypoints, 5 janvier 2004. [www.cs.ubc.ca/~lowe/papers/ijcv04.pdf](http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf). Consulté le 4 août 2014.
- [8] Jean-Marc Molina. Technique 3d temps réel, 1er décembre 1997. [http://goahomepage.free.fr/iut/carte\\_2d\\_3d/index.html](http://goahomepage.free.fr/iut/carte_2d_3d/index.html). Consulté le 20 août 2014.
- [9] Nicolas Thome. Calcul de descripteurs locaux et de dictionnaires visuels, février 2014. <http://webia.lip6.fr/~thomen/Teaching/INDIX/TME-SIFT-KMeans.pdf>. Consulté le 4 août 2014.

- [10] Wikipedia. Détection de zones d'intérêt. [http://fr.wikipedia.org/wiki/D%C3%A9tection\\_de\\_zones\\_d%27int%C3%A9r%C3%AAt](http://fr.wikipedia.org/wiki/D%C3%A9tection_de_zones_d%27int%C3%A9r%C3%AAt). Consulté le 17 juillet 2014.
- [11] Wikipedia. Photogrammétrie. <http://fr.wikipedia.org/wiki/Photogramm%C3%A9trie>. Consulté le 15 juillet 2014.
- [12] Wikipedia. Scale-invariant feature transform. [http://fr.wikipedia.org/wiki/Scale-invariant\\_feature\\_transform](http://fr.wikipedia.org/wiki/Scale-invariant_feature_transform). Consulté le 21 juillet 2014.