



FACULTÉ D'ÉLECTRONIQUE ET D'INFORMATIQUE

DÉPARTEMENT D'INFORMATIQUE

*Projet de vision par ordinateur*  
***Reconstruction 3D photométrique***

Réalisé par :

- DJEBROUNI Radhia (Groupe: 2)
- ABDELALI Asma Nihad (Groupe: 1)

Responsable du module : L. Abada

# Introduction

La reconstitution photométrique, autrement appelée la *Photometric Stereo*, est une technique qui permet de récupérer les informations d'une surface ou objet en utilisant plusieurs photos du même angle mais en utilisant plusieurs sources de lumière tour à tour. Les variations dans les réflexions de lumière augmentent l'information disponible sur l'objet, ce qui nous donne un avantage sur les images uniques avec une seule source de lumière.

Pour chaque point de la surface de notre objet, on utilise la direction des lumières et leurs intensités pour définir les positions possibles que prend ce point dans l'espace.

## Étape 1. Préparation de données

L'objectif de cette étape c'est de récupérer les informations du dataset, soit:

- Les positions des sources lumineuses dans une matrice, les positions sont sur trois composantes x, y et z.
- Les intensités des sources lumineuses qui représentent l'intensité de chaque pixel sous format R, G, B
- Le masque de l'objet, une image binaire qui nous permet de diminuer le temps d'exécution du programme ou on va ignorer les pixels qui sont noirs sur le masque soit ont une valeur de 0.
- Les images elles-mêmes.

## Récupération des images du dataset:

Les images dont les noms de fichier se trouvent dans le fichier "filenames.txt" et qui représentent chacune une photo de notre objet à reconstituer à partir d'un appareil photo fixe mais chacune avec un point de lumière différemment positionné ainsi qu'avec une intensité propre à lui.

On charge nos images en utilisant le type float pour éviter la perte de données, les images seront donc encodées sur 16 bits donc on divise la valeur de chaque pixel sur  $2^{16} - 1$  pour les normaliser et obtenir des valeurs comprises entre 0 et 1.

Afin de calculer les niveaux de gris, on divise chaque pixel par l'intensité de la source lumineuse qui a été utilisée pour prendre la photo en particulier.

A la fin de ces étapes on redimensionne la matrice représentant l'image pour que l'image soit sur une seule ligne, on ajoute cette ligne dans une matrice contenant toutes les images traitées. Une colonne de cette matrice représente le même pixel dans toutes les images.

## Étape 2. Calcul des normales

Un vecteur normal est le vecteur perpendiculaire à la tangente d'une surface, en calculant les normales de l'image on peut connaître la direction de chaque pixel de l'image en niveaux de gris.

Chaque pixel de l'objet 3D en niveaux de gris est représenté par le produit de la direction de la source lumineuse par son vecteur normal. De façon que :

$$E = S * N$$

Avec E l'éclairement d'un point de l'objet 3D, S le vecteur représentant la direction de la source lumineuse et N le vecteur normal.

Avec E et S connus nous obtenons l'équation:

$$N = S^{-1} * E$$

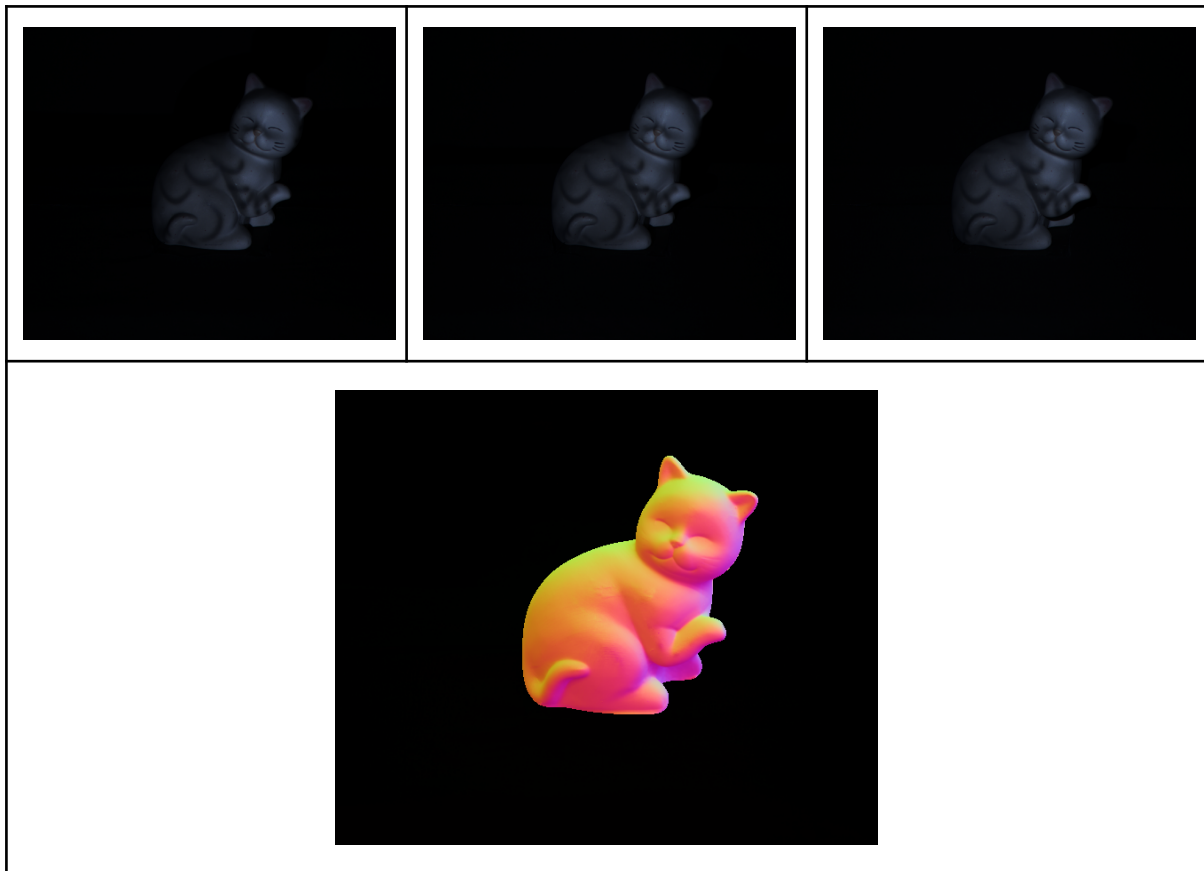
$S^{-1}$  Représente la matrice pseudo inverse de la matrice des directions des sources de lumière, E est une colonne de la matrice des images en niveaux de gris (Un pixel) et N sera un seul vecteur normal.

La matrice des vecteurs normaux est obtenue en faisant le produit de la matrice des images. Le résultat est ensuite remodelé et découpé par la largeur du masque (ou l'une des images d'entrée) afin de reformer une image. Ensuite, on normalise chaque pixel en utilisant l'équation suivante :

$$N = ( Nx/\| N\| , Ny/\| N\| , Nz /\| N\| ) \text{ tel que } N = \sqrt{( Nx^2 + Ny^2 + Nz^2)}$$

Chaque pixel n'est gardé que si le pixel équivalent dans le masque est blanc (valeur 1), afin de minimiser le temps d'exécution.

Résultat de cette étape :



### Étape 3. Intégration des normales pour déterminer Z

Pour cette étape nous avons testé deux approches dont seulement l'une d'entre elles (la deuxième) a donné un résultat acceptable, la première approche a consisté à calculer les vecteurs Z en utilisant l'angle entre la tangente et le vecteur X, dans la deuxième méthode nous avons utilisé les vecteurs p et q en suivant l'équation vue en cours.

#### 1- Calcul de Z en utilisant la translation des vecteurs normaux :

Avec cette méthode nous avons calculé les tangentes en calculant les vecteurs perpendiculaires aux normaux, puis on a calculé les angles entre chaque vecteur Z et le vecteur X [1,0,0].

La profondeur est alors calculé en utilisant la tangente de l'angle trouvé où:

$$profondeur = \frac{\tan(\alpha)}{|X|} \text{ avec } |X| \text{ longueur du vecteur X}$$

en sachant que la longueur du vecteur X est égale à 1.

#### 2- Calcul de Z en utilisant la méthode traitée en cours:

Cette étape est effectuée après avoir calculer p et q : les gradients par rapport aux axes x et y respectivement c-à-d la différence de Z entre deux points (pixel) de l'objet , en utilisant les équations suivantes :

$$N_x = -p/\sqrt{(q^2 + p^2 + 1^2)}, N_y = -q/\sqrt{(q^2 + p^2 + 1^2)}, N_z = 1/\sqrt{(q^2 + p^2 + 1^2)},$$

d'où on va trouver :  $p = -N_x/N_z$ ,  $q = -N_y/N_z$

D'où on construit la table z, sachant qu'on a la différence entre les valeurs z entre chaque pixel et pixel, on utilise les deux boucles suivantes :

```
z[0][0]=0
```

- calculer la profondeur z pour la première colonne

```
for y in range(1,w-1):
```

```
    z[0][y]=z[0][y-1] - q[0][y]
```

- calculer la profondeur z pour chaque ligne

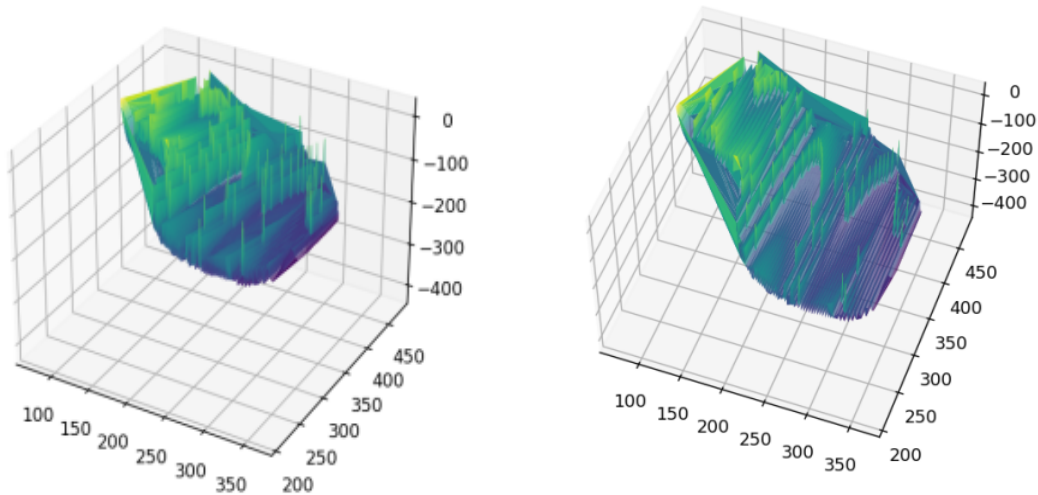
```
for x in range(1,h-1):
```

```
    for y in range(1,w-1):
```

```
        z[x][y]=z[x-1][y] - p[x][y]
```

### Affichage 3d :

En utilisant la librairie matplotlib, on a affiché les résultats obtenus dans les deux approches traitées pour calculer la profondeur z en utilisant la méthode trisurf après avoir échoué à faire fonctionner la méthode surface plus adéquate à répondre à l'énoncé, voici le résultat obtenu :



### **Conclusion :**

Dans la première et la deuxième étape de ce projet, nous avons pu effectuer la reconstruction 3D d'une image 2D en utilisant l'image 2D capturée 96 fois selon la position de la source lumineuse. Dans la troisième étape, on traité deux approches différentes afin de trouver la forme 3D finale, après avoir vu les résultats, on peut conclure que la deuxième méthode et la plus proche du résultat demandé .