

# TP photogrammétrie et construction d'un Modèle Numérique de Terrain

Constance Bau

Février 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Des photos au nuage de points</b>	<b>2</b>
2.1	Obtention des coins du damnier . . . . .	2
2.2	Reconstitution du nuage de points 3D . . . . .	6
<b>3</b>	<b>Du nuage de points au MNT volumique</b>	<b>8</b>
<b>4</b>	<b>Conclusion</b>	<b>10</b>

## 1 Introduction

Des marins pleins d'imagination aux modélisations mathématiques et simulations numériques il n'y a qu'un pas, ou peut être un peu plus. Voyons le chemin de l'un à l'autre. Depuis la mer, des marins imaginatifs crurent voir des vaches noires sur la plage, étonnant non ? Du moins assez pour qu'ils en parlent et que cet endroit, où étaient tombés des blocs de pierres des falaises avoisinantes, qui, recouverts d'algues, pourraient faire penser à des vaches noires, prenne le nom de Falaises des Vaches Noires. Ces fameuses falaises, hautes de plus de 100 mètres, s'étendant sur cinq kilomètres entre Villers-sur-Mer et Houlgate et site de découvertes paléontologiques majeures datant du jurassique, sont sujettes à l'érosion. Des physiciens, mathématiciens, grimpeurs, ...(tout aussi imaginatifs que les marins) ont alors été chargés, dans le cadre du projet TélÉDEtac, d'étudier l'évolution du trait de côte pour mieux comprendre, à l'échelle locale, les phénomènes d'érosion des falaises. Pour cela, ils ont du utiliser de nombreuses méthodes mathématiques et physique comme, par exemple, la photogramétrie, les mesures de résistivité électrique, l'algorithme de Gauss-Newton, ... Tentons de résumer brièvement les étapes de ce projet.

Tout d'abord, à partir de 40 photos haute résolution du trait de côte prises par drone, les coordonnées de millions de points ont pu être déduites grâce à

la photogrammétrie. Ensuite, un maillage 3D des falaises à pu être construit à partir de ces coordonnées. Finalement, les mesures de résistivité physiques et beaucoup de résolutions d'équations sur les noeuds du maillage ont permis de modéliser la structure géologique de ces falaises. Cela peut ensuite permettre de suivre l'érosion du littoral et de mieux comprendre les phénomènes d'érosion.

Quant à nous, nous allons regarder d'un peu plus près la première étape de ce projet et tenter de comprendre comment passer, grâce, entre autre, à la photogrammétrie, d'une photo aérienne à un maillage.

## 2 Des photos au nuage de points

Dans cette partie, à partir de 2 images représentant chacune une prise de vue différente d'une même scène nous allons reconstruire un nuage de point représentant la relief, la forme de cette scène.

### 2.1 Obtention des coins du damnier

A partir de deux prises de vue par drone de la même scène, deux images en damnier déformé (voir images ci-dessous) ont été créées. L'opérateur de détection de coins Harris est un opérateur capable de détecter les coins des cases du damier grâce à des calculs de gradients. Le fichier `traitementImg.py`, dans lequel est implémenter l'opérateur Harris, grâce à des calcul de dérivée d'image, prend en entrée une image en damier déformé. A partir de cela, il crée une image représentant seulement les coins des case du damier (voir Figure 5). De plus, et c'est surtout cela qui nous sera utile par la suite, il crée un fichier (`PtImg1.data` par exemple) contenant les coordonnées de ces coins. A partir de nos deux images en damier nous avons donc obtenu 2 fichiers de coordonnées de points. A partir de ces deux fichiers de coordonnées en 2D provenant de photo depuis différents points de vue, nous allons pouvoir reconstruire le nuage 3D de point représentant le relief de la scène prise en photo.

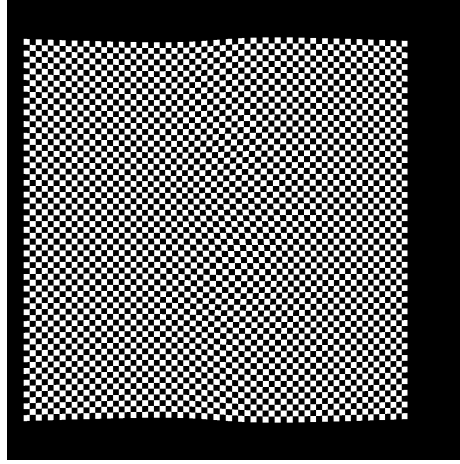


Figure 1: Photos de synthèse de la scène 3D prise de vue du point  $C_1$  .

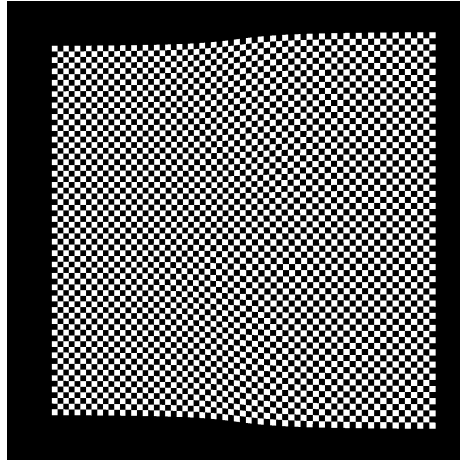


Figure 2: Photos de synthèse de la scène 3D prise de vue du point  $C_2$ .

Ces images correspondent à la prise de vue d'une même scène 3D de deux points de vue différents, comme illustré Figure 3, avec les caractéristiques suivantes :

- la position  $C_1 = (5, 10, 40)$ , la focale  $F = (0, 0, -0.08)$  et l'orientation de l'appareil  $(\theta_x, \theta_y, \theta_z) = (0, \frac{\pi}{20}, 0)$ ,
- la position  $C_2 = (10, 10, 40)$ , la focale  $F$  et l'orientation de l'appareil  $(\theta_x, \theta_y, \theta_z) = (0, 0, 0)$ .

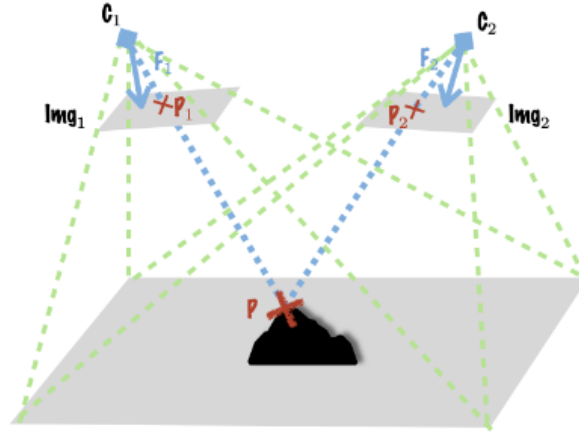


Figure 3: Illustration de la méthode de photogrammétrie.

La Figure 4 permet de mieux comprendre ces paramètres, en particulier l'orientation de la prise de vue. Cette description (simple) de la prise de vue correspond à un modèle chambre noire de l'appareil photo.

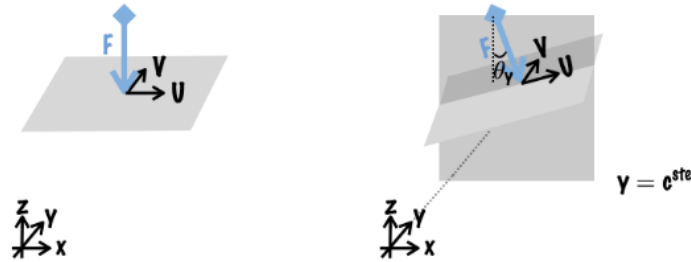


Figure 4: Modélisation de la prise de vue, à gauche un cas sans rotation, à droite avec une rotation  $\Theta_y$  dans le plan  $(Oxz)$ .

#### Idée générale du fonctionnement de l'opérateur Harris

Avant tout chose il faut bien avoir en tête qu'un coin est un point où la couleur change dans plusieurs directions en même temps. L'idée est donc de calculer les changements de couleur dans différentes directions grâce aux gradients, puis de considérer comme coins les points pour lesquels des changements de couleur brutaux apparaissent dans différentes directions perpendiculaires. Voyons comment faire cela en pratique.

Soit  $I(x, y)$  l'intensité de l'image en un point  $(x, y)$ . On considère une petite région  $W$  autour de ce point et on analyse comment cette région change lorsqu'on la décale légèrement de  $(\Delta x, \Delta y)$ .

On mesure cette variation avec la somme des différences au carré (SSD) :

$$f(\Delta x, \Delta y) = \sum_{(x_k, y_k) \in W} (I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$

Grâce au développement de Taylor, on peut approximer ce changement en fonction des dérivées partielles  $I_x$  et  $I_y$  de l'image :

$$f(\Delta x, \Delta y) \approx \sum_{(x, y) \in W} (I_x(x, y)\Delta x + I_y(x, y)\Delta y)^2$$

On peut réécrire cette expression sous forme matricielle :

$$f(\Delta x, \Delta y) \approx \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}^T M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

avec  $M$ , la matrice structurelle :

$$M = \sum_{(x, y) \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Pour savoir si c'est un coin, l'idée est d'étudier les valeurs propres  $\lambda_1$  et  $\lambda_2$  de  $M$  :

- Si une des valeurs propres est grande et l'autre petite  $\rightarrow$  c'est un bord.
- Si les deux valeurs propres sont grandes  $\rightarrow$  c'est un coin.
- Si les deux valeurs propres sont petites  $\rightarrow$  ce n'est ni un coin, ni un bord.

On introduit alors un score de Harris pour classifier les pixels :

$$R = \det(M) - k \cdot \text{trace}(M)^2$$

avec  $k$  un paramètre empirique ( $k \approx 0.04$ ).

- $R > 0 \rightarrow$  coin
- $R < 0 \rightarrow$  bord
- $R \approx 0 \rightarrow$  région plate

Finalement, pour éviter de détecter trop de coins proches, on garde uniquement ceux avec le score  $R$  le plus élevé localement.

### Exécution du programme traitementImg.py

En exécutant le programme traitementImg.py, on obtient le fichier img1Corner.png (figure 5) qui représente uniquement les coins du damier de l'image img1.png ainsi que le fichier img2Corner.png (figure 6) qui représente uniquement les

coins du damier img2.png. Les coordonnées de ces coins sont enregistrées dans les fichiers PtIm1.data et PtIM2.data respectivement.

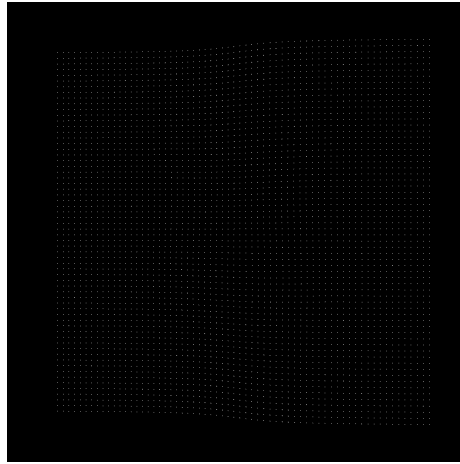


Figure 5: Image img1Corner.png ne représentant que les coins du damier déformé

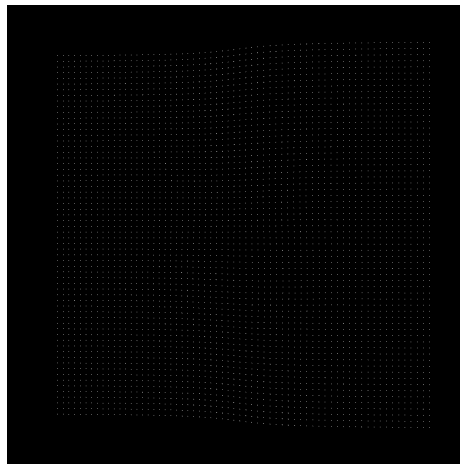


Figure 6: Image img2Corner.png ne représentant que les coins du damier déformé

## 2.2 Reconstitution du nuage de points 3D

Voyons maintenant comment reconstruire un nuage de points 3D à partir des deux fichiers de coordonnées des coins du damier, de la position de l'appareil photo, de la focale et de l'orientation de l'appareil photo. Pour cela, il faut

d'abord obtenir les coordonnées des coins du damier dans le repère  $(Oxyz)$  puis nous devons résoudre un système pour trouver les coordonnées des points du nuage de points.

**Changement de repère:** Étant donnée une prise de vue de caractéristiques  $C$ ,  $F$  et d'orientation  $(\theta_x, \theta_y, \theta_z)$ , et les coordonnées  $\tilde{p}$  d'un point sur l'image dans le repère  $(U, V)$ , nous allons les coordonnées de ce même point  $p$  dans le repère  $(Oxyz)$ .

Dans notre cas, on a  $(\theta_x, \theta_y, \theta_z) = (0, \theta_y, 0)$ , le repère  $(U, V)$  est donc en rotation selon  $Y$  par rapport au repère  $(Oxyz)$

Nous définissons alors la matrice de rotation autour de  $Y$  de la manière suivante :

$$R_Y = \begin{pmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{pmatrix}$$

On applique alors une rotation et une translation au point  $\tilde{p} = (\tilde{p}_u, \tilde{p}_v, 0)$  :

$$p = C + F' + R_Y \cdot \tilde{p}$$

où  $F' = R_Y \cdot F$ .

Maintenant, voyons le système à résoudre pour obtenir les coordonnées d'un point  $P$  dans le domaine 3D à partir de ses coordonnées  $P1$  et  $P2$  sur deux prises de vue différentes. On rappelle qu'on note  $C1$  et  $C2$  les positions des caméras ayant pris ces deux prises de vue.

La photogrammétrie repose sur le principe suivant : l'appareil photo, le point sur la photo et la point dans la réalité sont alignés. Donc  $C1, P1$  et  $P$  sont alignés. Et de même  $C2, P2$  et  $P$ .

A partir de cela on obtient donc les équations suivantes :

$$\begin{cases} (P - C1) = \alpha_1(P1 - C1) \\ (P - C2) = \alpha_2(P2 - C2) \end{cases}$$

ce qui nous donne :

$$\begin{cases} P - \alpha_1(P1 - C1) = C1 \\ P - \alpha_2(P2 - C2) = C2 \end{cases}$$

On met cela sous forme d'un système linéaire

$$AX = B$$

que doit satisfaire le point  $P$ , avec

$$A \in M_{6,5}, \quad X \in \mathbb{R}^5, \quad B \in \mathbb{R}^6$$

tels que :

$$A = \begin{bmatrix} Id_3 & C1 - P1 & 0 \\ Id_3 & 0 & C2 - P2 \end{bmatrix}, \quad B = \begin{bmatrix} C1 \\ C2 \end{bmatrix} e$$

où

$$X = \begin{bmatrix} X \\ Y \\ Z \\ \alpha_1 \\ \alpha_2 \end{bmatrix}$$

Le système obtenu possède plus de coefficient connus qu'inconnus, il est donc sur-déterminé. Pour le résoudre on préconise alors d'utiliser la méthode des moindres carrés : on cherche donc  $X = (x, y, z, \alpha_1, \alpha_2)$  qui minimise

$$\|AX - B\|_2$$

En python, on prend alors la méthode `np.linalg.lstsq`.

Après exécution du code implémentant ces procédés en python, on obtient le nuage de points suivant :

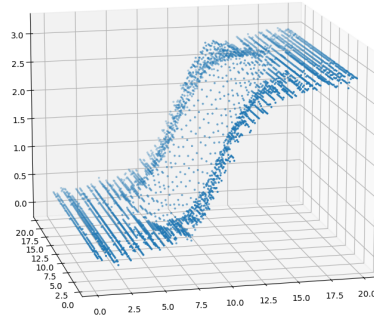


Figure 7: Nuage de points obtenu

### 3 Du nuage de points au MNT volumique

Maintenant que nous avons le nuage de points, dans la partie qui suit, nous allons voir comment construire un maillage volumique 3D dont un bord correspond à



la topographie des falaises représentée par notre nuage de points. Pour cela, nous allons construire un maillage sur un cube et modifier l'une de ses faces selon la topographie de la falaise.

Voici la méthode utilisée :

nous allons déformer un cube  $(0, 1)^3$  maillé de sorte que la face  $(0, 1)^2 \times \{z = 1\}$  du cube devienne la surface du sol. Dans le cas où la surface est décrite par une fonction  $z = f(x, y)$ , la technique consiste simplement à changer la coordonnée  $z$  d'un point du cube de coordonnées  $(x_i, y_i, z_i)$  comme suit :

$$Z_i = z_m(1 - z_i) + z_i f(x_i, y_i), \quad (1)$$

où  $z_m < \min f(x, y)$  est un paramètre décrivant la profondeur maximale qu'on souhaite considérer.

Pour généraliser au cas où l'on ne connaît pas la fonction  $f$  mais uniquement le nuage de points, nous allons faire de l'interpolation.

Nous allons utiliser 2 méthodes d'interpolation, à savoir l'interpolation du plus proche voisin et une interpolation linéaire et nous comparerons les maillages obtenus.

#### **Méthode 1 : Interpolation du plus proche voisin :**

Pour réaliser cette méthode, nous remplaçons l'évaluation de  $f(x_i, y_i)$  par  $z_p$  où  $p$  est l'indice du point du nuage de points de coordonnées  $(x_p, y_p)$  le plus proche de  $(x_i, y_i)$ , coordonnées d'un noeud du maillage initiale sur la surface du cube que déforme. Elle produit des résultats discontinus avec des transitions abruptes.

#### **Méthode 2 : Interpolation linéaire :**

Pour chaque points  $p_i$  de coordonnées  $(x_i, y_i)$  sur la face à déformer du cube, nous cherchons ses 3 plus proches voisins  $k_1 = (x_{i,1}, y_{i,1}, z_{i,1})$ ,  $k_2 = (x_{i,2}, y_{i,2}, z_{i,2})$  et  $k_3 = (x_{i,3}, y_{i,3}, z_{i,3})$  dans le nuage de points. Ces 3 points deviennent nos 3 points d'interpolation. Puis nous définissons les poids de l'interpolation en fonction de la distance des voisins. C'est à dire que plus un point voisin est proche du point pour lequel on fait l'interpolation, plus celui-ci "pèsera lourd" dans l'interpolation. On prend alors :

$$w'_1 = \frac{1}{\text{distance}(p, k_1)}, w'_2 = \frac{1}{\text{distance}(p, k_2)} \text{ et } w'_3 = \frac{1}{\text{distance}(p, k_3)}$$

$$\text{où } \text{distance}(p, k_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$\text{De plus on } w_{tot} = w'_1 + w'_2 + w'_3$$

$$\text{On normalise les poids par } w_{tot} : w_i = \frac{w'_i}{w_{tot}}.$$

A partir de cette interpolation on modifie alors les points du maillage de la manière suivante : le point  $p = (x_i, y_i, z_i = 1)$  est transformé en  $\tilde{p} = (x_i, y_i, z_i = w_1 z_{k_1} + w_2 z_{k_2} + w_3 z_{k_3})$ . En combinant les valeurs des points voisins pour créer des transitions fluides et des surfaces régulières, cette méthode présente une meilleure précision et continuité mais elle est légèrement plus complexe à calculer.

Voyons les maillages obtenus :

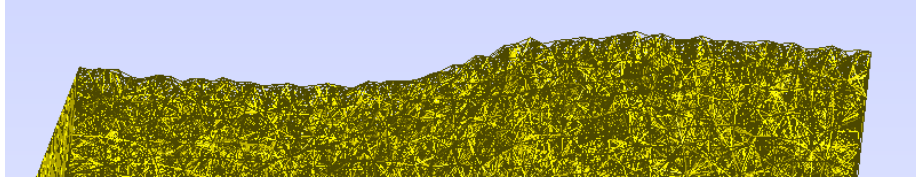


Figure 8: surface du sol obtenue avec interpolation du plus proche voisin

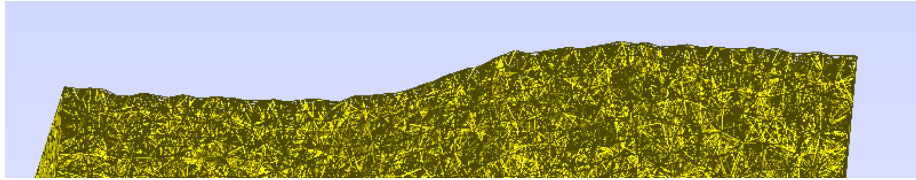


Figure 9: surface du sol obtenue avec interpolation linéaire

Les différences sont assez discrètes mais l'on peut remarquer que les pentes sont plus abruptes et les sommets plus pointus avec l'interpolation du plus proche voisin qu'avec l'interpolation linéaire. La tranche de surface que l'on peut voir est beaucoup plus anguleuse avec la technique du plus proche voisin.

Avec l'interpolation linéaire, les angles sont adoucis et les formes courbées. On pourrait comparer la tranche de surface obtenue par le plus proche voisin à des montagnes jeunes et la même tranche de surface mais cette fois obtenue avec une interpolation linéaire à des montagnes anciennes.

Néanmoins, lorsque l'on voit des photos des falaises du site des Vaches Noires, on remarque que les falaises sont plutôt tranchantes. L'interpolation linéaire n'était donc peut-être pas nécessaire, pour représenter ce type de surface, l'interpolation du plus proche voisin aurait pu suffire.

## 4 Conclusion

Durant ce travail, nous avons pu mettre en pratique différentes méthodes mathématiques pour modéliser un problème concret. Nous avons pu réaliser les différentes étapes permettant de passer des photos d'un site géologique à un maillage de ce site. Le temps d'exécution de nos codes était raisonnable mais l'on peut imaginer que le temps d'exécution peut rapidement augmenter pour des sites plus complexes ou plus grands et des mailles plus fines. On a aussi pu voir 2 types d'interpolation. Dans notre cas, l'interpolation du plus proche voisin aurait pu suffire car les structures géologiques étudiées sont plutôt anguleuses et tranchantes. Dans d'autres cas avec des structures plus continues, comme par exemple de vieilles montagnes, une interpolation linéaire serait plus adaptée pour ne pas perdre les arrondis de la structure.