**Travail demandé.** Vous utiliserez de préférence les langages `Python` ou `Matlab`.

1. *Paires stéréoscopiques.* L'objet considéré est un polyèdre défini par des points $P_1, \ldots, P_n$ reliés entre eux par des arêtes. Ecrire un script pour calculer les transformations $\mathcal{T}_G$ et $\mathcal{T}_G$ de ce polyèdre et pour afficher les deux images (une pour chaque oeil) sur un même graphique. Tester votre code avec les polyèdres suivants.

   (a) *Tétraèdre :*
$$P_1 = (1.2, -0.5, 65), \quad P_2 = (3, 1.5, 63),$$
$$P_3 = (4, -1, 62), \quad P_4 = (1.8, -1.5, 62)$$

   (b) *Prisme :*
$$P_1 = (1, -1, 65), \quad P_2 = (1.8, -2, 62), \quad P_3 = (4.2, -1.5, 65),$$
$$P_4 = (1, 2, 65), \quad P_5 = (1.8, 1, 62), \quad P_6 = (4.2, 1.5, 65)$$

   (c) *et encore un polyèdre... :*
$$P_1 = (0, 2, z), \quad P_2 = (0, 3, z), \quad P_3 = (3, 3, z), \quad P_4 = (3, 2, z),$$
$$P_5 = (2, 2, z), \quad P_6 = (2, 0, z), \quad P_7 = (1, 0, z), \quad P_8 = (1, 2, z),$$

   puis 8 autres points avec $P_{8+i} = P_i + (0, 0, 1)$, $i = 1, \ldots 8$. Prendre $z = 64$.

   On prendra la distance entre les yeux $e = 7cm$ et la distance des yeux au plan de projection $f = 40cm$.

   Prendre d'autres objets polyédriques plus ou moins compliqués. Vérifier que si la taille des images est plus grande que $e/2$ alors l'effet stéréoscopique disparait ou s'estompe fortement. Faire varier aussi la distance $f$.

2. *Cas d'une surface : points aléatoires.* L'objet considéré est défini par une surface $z = h(x, y)$. Engendrer *aléatoirement* des points sur cette surface, calculer leurs projections avec (1) et afficher le nuage de points ainsi obtenu dans le plan de projection. Vous pourrez considérer le "pont parabolique" suivant :

$$z = h(x, y) = \begin{cases} \dfrac{x^2}{4} - \dfrac{3x}{2} + 62 & \text{si } |y| \leq 8 \text{ et } -6 \leq x \leq 12 \\ 80 & \text{sinon.} \end{cases} \tag{8}$$

3. *Cas d'une surface : calculs des points sur les courbes* $(C)$. La surface $z = h(x, y)$ est définie pour $(x, y) \in [a, b] \times [c, d]$. On segmente l'intervalle $[c, d]$ (pour $y$) en plusieurs coupes $y = y_i$ pour $i = 0, \ldots, m$ avec $c = y_0 < y_1 < \cdots < y_m = d$ et les $y_i$ étant espacés régulièrement. Pour chaque courbe $(C_i)$ définie par $z = h(x, y_i)$, $i = 0, \ldots, m$, vous devez :
   - engendrer aléatoirement plusieurs points $P$.
   - pour chacun des points $P$, déterminer les points $I_0^P, I_1^P, \ldots, I_n^P$ dont les abscisses sont dans l'intervalle $[a, b]$. On s'arrêtera quand l'abscisse de $I_n^P$ est très proche de $b$. Le nombre $n$ de points dépend de $i$ et du point de départ $P$.
   - afficher tous les points $I_j^P$ ainsi déterminés pour obtenir le stéréogramme de la surface.

   Le calcul des points $I_j^P$ doit se faire en utilisant les 3 procédures différentes décrites précédemment (cf. Calcul de $I_2$). Tester avec la surface (8), comparer les 3 procédures entre-elles et aussi avec la procédure précédente du choix aléatoires des points.

   Ensuite, à vous de jouer ... pour trouver des objets et surfaces intéressants, pour ajouter de la couleur, remplacer les points par des *motifs esthétiques*[1] ...

## Références


[1] Dalang R., Chaabouni A., Algèbre linéaire, aide-mémoires, exercices et applications, PPUR, 2004.

[2] Quarteroni A., Sacco R., Saleri F., Méthodes numériques pour le calcul scientifique, Springer, 2006.

[3] Terrell M., Terrell R., *Behind the Scenes of a Random Dot Stereogram*, The American Mathematical Monthly , Oct., 1994, Vol. 101, No. 8, pp. 715-724


1. Dans la méthode présentée ici, le motif de base est le point (chaque point $P$ donne lieu à une suite de points $I_0, I_1, I_2, \ldots$). Il est possible de remplacer les points $P$ par des motifs $M$ (cercle, croix, courbe, figure géométrique, ...). Le principe reste le même avec un motif $M$ : on calcule $J_1 = \mathcal{T}_D(M)$ l'image du motif $M$ par $\mathcal{T}_D$ puis le motif $M_1 = \mathcal{T}_G^{-1}(J_1)$ et ainsi de suite ...