TP5 – RTI (*Reflectance Transformation Imaging*)

Exercice 1 : étalonnage des éclairages

L'étalonnage de l'éclairage est effectué à partir d'images telles que celle de la figure 1 : la direction de l'éclairage se déduit de la position du point brillant sur chacune des deux sphères noires.

Figure 1 – Chacune des sphères noires donne une indication sur la direction de l'éclairage. En moyennant ces estimations, on obtient une assez bonne estimation de l'éclairage au niveau de l'objet d'intérêt. Les silhouettes des sphères sont inscrites dans les carrés de couleur (l'ombre de la sphère peut prêter à confusion dans la délimitation des silhouettes).

Le fichier `spheres.mat` contient m photographies de chaque sphère noire, concaténées dans la matrice `spheres`, de taille nb_lignes $\times$ nb_colonnes $\times$ m $\times$ nb_spheres. La matrice `spheres(:,:,i,k)` contient la sous-image de la k-ème sphère sous l'éclairage numéro i. Ces images étant bruitées, il convient de leur appliquer un filtre avant de rechercher le point le plus brillant, en utilisant la séquence suivante :

```
taille_noyau = round(rayon/20);
image_filtree = conv2(triangle(taille_noyau),triangle(taille_noyau),image_sphere,'same');
```

où la fonction `triangle` est définie par :

```
triangle = @(n)(1+n-abs(-n:n))/n;
```

Une fois le point le plus brillant détecté, sa position à l'intérieur de la silhouette de la sphère permet effectivement de connaître la direction de l'éclairage.

Écrivez la fonction `etalonnage`, appelée par le script `exercice_1`, qui doit retourner les m vecteurs $\mathbf{s}_i \in \mathbb{R}^3$, $i \in \{1,\ldots,m\}$, correspondant aux différents éclairages : chaque vecteur $\mathbf{s}_i$, de norme 1, décrit la direction moyenne du i-ème éclairage, égale à la moyenne des directions fournies par les différentes sphères noires.

Écrivez ensuite la fonction `conversion`, appelée par le script `exercice_1`, qui déduit de chaque vecteur $\mathbf{s}_i$ un couple d'angles $(\theta_i, \phi_i) \in [0, \pi/2] \times [-\pi, \pi[$ décrivant une des directions d'éclairage, et affiche ces angles dans un repère 2D (θ en abscisse, ϕ en ordonnée).

Exercice 2 : RTI sans reconstruction 3D

Le pixel de numéro $j \in \{1, \dots, n\}$ est caractérisé par m niveaux de gris $\{I_1^j, I_2^j, \dots, I_m^j\}$, qui correspondent aux m éclairages estimés par la procédure précédente. Pour pouvoir simuler un éclairage de direction quelconque, c'est-à-dire associer à chaque pixel un niveau de gris découlant d'une direction d'éclairage quelconque, définie par un couple d'angles $(\theta, \phi) \in [0, \pi/2] \times [-\pi, \pi[$, il est nécessaire d'interpoler les m triplets de valeurs $(\theta_i, \phi_i, I_i^j)$. Cela peut être réalisé, par exemple, à l'aide d'un *spline d'interpolation* polynomial.

Pour éviter le phénomène de *sur-apprentissage*, qui se traduit par des oscillations parasites de la fonction d'interpolation, il est nécessaire de limiter le nombre de paramètres. Nous choisissons d'utiliser une fonction polynomiale de degré 2, qui comporte seulement six paramètres $(a_{0,0}^j, a_{1,0}^j, a_{0,1}^j, a_{2,0}^j, a_{1,1}^j, a_{0,2}^j)$:

$$I^j(\theta, \phi) = a_{0,0}^j + a_{1,0}^j\,\theta + a_{0,1}^j\,\phi + a_{2,0}^j\,\theta^2 + a_{1,1}^j\,\theta\,\phi + a_{0,2}^j\,\phi^2 \tag{1}$$

qui doivent vérifier les m équations suivantes :

$$I_i^j = a_{0,0}^j + a_{1,0}^j\,\theta_i + a_{0,1}^j\,\phi_i + a_{2,0}^j\,\theta_i^2 + a_{1,1}^j\,\theta_i\,\phi_i + a_{0,2}^j\,\phi_i^2, \quad i \in \{1, \dots, m\} \tag{2}$$

Ces équations étant linéaires vis-à-vis des paramètres, nous pouvons les réécrire sous forme matricielle :

$$\begin{bmatrix} I_1^j \\ I_2^j \\ \vdots \\ I_m^j \end{bmatrix} = \begin{bmatrix} 1 & \theta_1 & \phi_1 & \theta_1^2 & \theta_1\,\phi_1 & \phi_1^2 \\ 1 & \theta_2 & \phi_2 & \theta_2^2 & \theta_2\,\phi_2 & \phi_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \theta_m & \phi_m & \theta_m^2 & \theta_m\,\phi_m & \phi_m^2 \end{bmatrix} \begin{bmatrix} a_{0,0}^j \\ a_{1,0}^j \\ a_{0,1}^j \\ a_{2,0}^j \\ a_{1,1}^j \\ a_{0,2}^j \end{bmatrix} \tag{3}$$

Nous pouvons regrouper les différents systèmes (3) correspondant aux n pixels :

$$\begin{bmatrix} I_1^1 & I_1^2 & \cdots & I_1^n \\ I_2^1 & I_2^2 & \cdots & I_2^n \\ \vdots & \vdots & \cdots & \vdots \\ I_m^1 & I_m^2 & \cdots & I_m^n \end{bmatrix} = \begin{bmatrix} 1 & \theta_1 & \phi_1 & \theta_1^2 & \theta_1\,\phi_1 & \phi_1^2 \\ 1 & \theta_2 & \phi_2 & \theta_2^2 & \theta_2\,\phi_2 & \phi_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \theta_m & \phi_m & \theta_m^2 & \theta_m\,\phi_m & \phi_m^2 \end{bmatrix} \begin{bmatrix} a_{0,0}^1 & a_{0,0}^2 & \cdots & a_{0,0}^n \\ a_{1,0}^1 & a_{1,0}^2 & \cdots & a_{1,0}^n \\ a_{0,1}^1 & a_{0,1}^2 & \cdots & a_{0,1}^n \\ a_{2,0}^1 & a_{2,0}^2 & \cdots & a_{2,0}^n \\ a_{1,1}^1 & a_{1,1}^2 & \cdots & a_{1,1}^n \\ a_{0,2}^1 & a_{0,2}^2 & \cdots & a_{0,2}^n \end{bmatrix} \tag{4}$$

La résolution de ce système en moindres carrés peut être effectuée à l'aide de l'opérateur \ de Matlab.

Écrivez la fonction `interpolation`, appelée par le script `exercice_2`, qui prend en entrée les m images concaténées dans la matrice `images` lue dans le fichier `donnees.mat` et les m couples d'angles (θ_i, ϕ_i) estimés précédemment, et qui retourne les paramètres des n fonctions d'interpolation I^j, $j \in \{1, \dots, n\}$. Ce script simule un éclairage tournant autour de l'axe optique, caractérisé par un angle θ_0 constant. Testez différentes valeurs de $\theta_0 \in [0, \pi/2]$. Pour quelle plage de valeurs de θ_0 la simulation est-elle réaliste ?

Exercice 3 : RTI avec reconstruction 3D

La reconstruction 3D à partir de données RTI nécessite de faire une hypothèse sur la surface de la scène 3D. Sous l'hypothèse lambertienne, caractéristique d'une surface mate, la normale au pixel numéro $j \in \{1, \dots, n\}$ pointe dans la direction de l'éclairage qui maximise le niveau de gris. Cela nécessite de résoudre le problème :

$$(\widehat{\theta}, \widehat{\phi}) = \underset{(\theta,\phi)}{\operatorname{argmax}}\, I^j(\theta, \phi) \tag{5}$$

où I^j désigne la fonction d'interpolation du niveau de gris du pixel numéro j. Néanmoins, comme le problème d'optimisation (5) doit être résolu sous les contraintes $\theta \in [0, \pi/2]$ et $\phi \in [-\pi, \pi[$, il n'est pas forcément simple de le résoudre. Une solution plus simple consiste à chercher le maximum des m niveaux de gris $\{I_1^j, I_2^j, \dots, I_m^j\}$.

Écrivez la fonction `normales`, appelée par le script `exercice_3`, permettant d'estimer les normales de la scène 3D observée, sous l'hypothèse lambertienne. Ce script affiche ensuite la reconstruction 3D obtenue par intégration du champ de normales ainsi estimé.