

Documentation

Par LORAIN Antoine et MARQUET Quentin

1. Fichier principal main.py

Le fichier main.py est le fichier principal du projet Pynect. Il contient les fonction `get_depth()` et `modifiedColors()` ainsi qu'une boucle principale. La fonction `get_depth()` permet de récupérer une image depuis le kinect à l'aide du module `freenect` (installation détaillée dans le fichier `documentation.docx` et `readme.md`) et converti les données en `np.array` sur module `numpy` grâce au fichier `frame_convert2`. La fonction `modifiedColors()` commence par convertir la `np.array` en tableau Python sous cette forme : `[[(255,255,255), (255,255,255)], [(255,255,255), (255,255,255)]]`

Chaque partie soulignée contient les valeurs R, V, B (Rouge, Vert, bleu) d'un pixel. Chaque couleur de surlignage correspond à une ligne de l'image.

Note : L'image est au format 640x480, il y a donc une liste de 480 éléments. Chaque liste contient 640 tuples qui contiennent chacun 3 valeurs.

2. Fichier sources/frame_convert2.py

Le fichier `frame_convert2.py` est un fichier fourni par `OpenKinect` (développeur de `libfreenect`). Il sert à convertir les données brutes de la kinect en une liste au format `numpy` qui sont plus faciles à traiter avec `python` et affichable avec le module `cv2`. Le fichier contient 3 fonctions mais seulement 2 sont utilisées dans notre cas. Nous appelons la fonction `pretty_depth_cv()` lors de la conversion des données dans le fichier `main.py`. `pretty_depth_cv` fait appel à la fonction `pretty_depth` qui elle s'occupe de convertir les données en `np.array`. Il est important de noter que le fichier `frame_convert2` génère une image en niveau de gris.

3. Fichier source/get_config_params.py

Afin d'avoir une meilleure personnalisation, nous avons mis en place un fichier `config.ini` qui permet de changer les couleurs utilisées par l'application. Le fichier `get_config_params.py` contient 3 fonctions. La fonction `create_parser()` qui est appelée au début des deux autres fonctions et qui retourne un objet du module `ConfigParser` qui contient les données de notre fichier `config.ini`. La fonction `get_colors()` prend en argument le chemin qui pointe sur le fichier `config.ini` en partant du dossier racine du `main.py`. La fonction `get_colors()` retourne une liste qui contient une liste qui contient elle-même un tuple avec la valeur de gris minimal et maximal et un autre tuple qui contient le code RVB de la couleur à appliquer.

La fonction `get_title()` retourne le titre défini dans le `config.ini` qui sera appliqué à la fenêtre qui affiche l'image en couleur. (voir `main.py`)

4. Fichier `config.ini`

Le fichier `config.ini` est lu par le fichier `get_config_params.py`. Un fichier `.ini` est organisé sous forme de sections qui peuvent accueillir les paramètres associés à une valeur. Pour une meilleure personnalisation, vous pouvez ajouter des sections sous cette forme ci :

```
[valeurMin,valeurMax]
```

```
colors=R,G,B
```

Où `valeurMin` et `valeurMax` sont en niveau de gris et R, G, B sont les codes RVB de la couleur à ajouter. Attention chaque section DOIT contenir le paramètre `colors`. Chaque valeur est séparée par une virgule sans espaces avant ni après.

Le titre de la fenêtre finale peut être modifié en changeant les paramètres `window_title` par une autre chaîne de caractères. Ce nouveau titre n'est pas obligatoirement entre guillemets.

5. Fichier `test/test_couleurs.py`

Le fichier `test_couleurs.py` est un fichier python appliquant le même algorithme que le fichier `main.py` mais sur l'image `image_de_test.png`. Ce fichier peut être utilisé pour tester une configuration de couleur comme vue dans le fichier `config.ini`.

6. Fichier `test/image_de_test.png`

Le fichier `image_de_test.png` est un fichier image contenant simplement un dégradé des valeurs de gris trouvable dans le bac à sable. Ce fichier est destiné à être utilisé comme fichier de test ou de démonstration lorsque vous n'avez pas de Kinect et/ou de bac à sable pour essayer le projet Pynect.