

TP1 : Prise en main de la bibliothèque DGtal

Représentation de volumes, parcours, traitement et affichage

Ricardo Uribe Lobello

27/11/2019

La bibliothèque DGtal est une bibliothèque constituée de classes templatées. Les templates offrent une plus grande souplesse à nos développements si nous arrivons à extraire des concepts qui sont communs à plusieurs types de données. Par exemple, l'opération plus (+) est supportée par tous les types de données numériques. Ainsi, un algorithme qui marche pour des entiers, devrait facilement être généralisable aux nombres réels.

Vous allez recevoir un code "startCode" de base de la part de votre intervenant. Ce code contient un fichier .cpp et un CMakeList.txt qui compile et génère l'exécutable du code. Le code utilise un viewer intégré en DGtal qui vous permettra de visualiser le résultat de vos traitements.

Partie A : Analyse et modification du code.

Le code de base fait le parcours d'un volume chargé. Le fichier à utiliser pour tester est "lobster.vol". Ce parcours illustre l'utilisation de la classe Z3i pour définir le domaine des objets volumiques et du template ImageContainer-BySTLVector pour gérer les voxels du volume.

1. Vous allez commencer par prendre quelques minutes pour analyser le code qui vous a été fourni afin de mieux comprendre la manière comme DGtal crée les classes à partir des templates et comment est-ce qu'elles sont instanciées. Exécutez le code et analysez la sortie.
2. Ensuite, essayez de modifier le code du parcours du volume pour que les voxels affichés soient ceux qui ont une valeur entre 100 et 150. Compilez et affichez le résultat. Que est-ce que vous voyez ? Qu'est-ce que cela vous indique ?
3. Modifiez le code pour afficher seulement les voxels pour lesquels la coordonnée x est plus grande qu'une valeur donnée. Qu'est-ce que vous observez ?
4. Maintenant, il faut extraire une sous image à partir de l'image courante. Pour cela, il faut déclarer une nouvelle image est un nouveau domaine. Les domaines sont définis en utilisant la coordonnées initiale et la coordonnée finale de la manière suivante.

```
Z3i::Point pointInitial (220, 50, 10);
Z3i::Point pointFinal (260, 100, 20);
Z3i::Domain sousDomain(pointInitial, pointFinal);
```

Utilisez ces instructions pour définir un domaine d'intérêt à l'intérieur de l'image courante. Affichez uniquement la région choisie.

5. Appliquez une coloration aux voxels basée sur une formule simple. Calculez la distance euclidienne au carré de chaque point du volume par rapport au centre du volume. Utilisez l'équation : $d_c(x, y, z) = (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2$ avec (x_c, y_c, z_c) étant le centre du domaine (qu'il faudra calculer). Il faudra définir une palette limitée des couleurs à utiliser. Elle est définie comme suit :

```
//Gradient de couleurs.
GradientColorMap<long> gradient(0,30);
gradient.addColor(Color::Red);
gradient.addColor(Color::Yellow);
```

```

gradient.addColor(Color::Green);
gradient.addColor(Color::Cyan);
gradient.addColor(Color::Blue);
gradient.addColor(Color::Magenta);
gradient.addColor(Color::Red);

// Extraction de la couleur à partir de la valeur de distance.
Color c= gradient(rangeValue);

//Affectation de la couleur à appliquer pour le prochain voxel à visualiser.
viewer << CustomColors3D( Color((float)(c.red()),
                                (float)(c.green()),
                                (float)(c.blue()),205)),
                        Color((float)(c.red()),
                                (float)(c.green()),
                                (float)(c.blue()),205));

```

Partie B : Visualisation et analyse de la transformée en distance implantée en DGtal

Dans cette partie, vous allez utiliser la librairie DGtal pour faire l'extraction de la transformée en distance 3D d'un modèle volumique. Dans le main du code donné en comme starter, appelez la fonction **transforméeEnDistance()** avec le volume "Al.100.vol". Ensuite, Réalisez les analyses et changements suivants.

1. Analysez le résultat obtenu. Est-ce qu'il semble logique par rapport à ce que vous avez appris concernant la transformée en distance ? Essayez de changer les paramètres utilisés pour la génération de la transformée, par exemple, le nombre d'échantillons de départ ou la manière comment est-ce qu'ils sont générés. Analysez les résultats.
2. Etudiez le code de la fonction **transforméeEnDistance()** en essayant de comprendre ce que la procédure fait et comment est-ce que ce type d'algorithmes sont implantés en DGtal.
3. Ensuite, utilisez la technique développée dans le point 3 de la partie A pour afficher seulement les voxels pour lesquels la coordonnée x est égale à une valeur donnée. Par exemple, le milieu du domain. Cela vous permettra d'observer le résultat de la transformée en distance sur un espace bidimensionnel.
4. Visualisez uniquement les voxels se trouvant dans un range de distance des germes donnés en paramètre à la transformée. Cela est équivalent à extraire les iso-surfaces discrètes de la fonction définie par la transformée en distance.
5. Réfléchissez à la manière comment la transformée en distance que vous venez d'étudier est relationnée avec l'extraction de la transformée en distance 3D classique concernant la distance de tous les voxels du volume au complémentaire de la forme. Comment est-ce qu'on pourrait obtenir ce résultat avec l'algorithme utilisé dans cet exemple.