

Graphs : une bibliothèque de manipulation de graphes géométriques sous Matlab

D. Legland

19 mai 2011

En traitement d'images ou en modélisation, on est souvent amené à manipuler des structures de données qui sont bien représentées sous la forme de graphe. J'ai développé plusieurs fonctions sous matlab, que j'ai regroupées dans une bibliothèque relativement autonome. Ce document présente les structures de données utilisées, ainsi que les différentes fonctions disponibles.

Table des matières

1	Introduction	3
1.1	Objectifs	3
1.2	Conventions	3
2	Représentation des données	3
2.1	Variables séparées	3
2.2	Structure de données	4
3	Création de graphes	4
3.1	Création simple	4
3.2	Graphes géométriques	4
3.3	Graphes d'une image	5
4	Manipulation de graphes	5
4.1	Extraction d'informations	5
4.2	Opérations de bas niveau	6
4.3	Opérations générales	7
5	Opérations particulières	8
5.1	Filtrage sur les sommets d'un graphe	8
5.2	Opérations sur les graphes géométriques	8
5.3	Diagrammes de Voronoï	9
6	Fonctions d'affichage	9
6.1	Affichage général	9
6.2	Affichage d'informations sur le graphe	10

1 Introduction

Un graphe est constitué d'un ensemble de sommets ou noeuds, et d'un ensemble d'arêtes qui décrivent les relations entre ces noeuds. J'appelle graphe géométrique un graphe dont les sommets sont des points du plan ou de l'espace (on peut généraliser à des points de \mathbb{R}^d , mais je n'en ai pas encore eu l'utilité).

1.1 Objectifs

La bibliothèque fournit des fonctions pour plusieurs types de problèmes :

- Génération de graphes (quelques modèles types, mais en fait la génération se fait le plus souvent à partir de données réelles)
- Modification de graphes existants : ajout et suppression de sommets ou d'arêtes, fusion de graphes, extraction de sous-graphes...
- Algorithmes classiques sur les graphes. Par exemple, test de planarité d'un graphe, colorabilité avec un nombre donné de couleurs, mesures telles que le diamètre, algorithme de Dijkstra...
- Caractérisation d'un graphe. Cette partie est peut-être plus spécifique. L'idée est de construire une application qui à un graphe donné associe une mesure, qui peut être un scalaire, un vecteur, une matrice...
- Opérations morphologiques sur des graphes valués
- Propagation de distances, et calcul de paramètres géodésiques (centre, diamètre...)

1.2 Conventions

Quelques conventions utilisées pour la bibliothèque :

- les noms des fonctions suivent la convention « Java », ex : `unNomDeFonction`
- la langue utilisée est l'anglais
- les noms des fonctions commencent pour la plupart par « gr » (ex : `grNeighbourNode`), ou alors contiennent le mot « graph » (ex : `clipGraph`).
- on passe les infos sur le graphe en premier, puis les infos complémentaires

2 Représentation des données

Un graphe peut être représenté de deux manière différentes. Soit en déclarant séparément les tableaux décrivant les sommets et les arêtes, soit en les regroupant dans une structure de données.

2.1 Variables séparées

Les différentes fonctions sur les graphes utilisent en général les deux premiers argument sous la forme :

nodes : un tableau contenant les coordonnées x et y (éventuellement z pour les graphes 3D) de chaque sommet. Chaque sommet est identifié par son indice, qui correspond à sa ligne dans le tableau des sommets.

edges : un tableau de 2 colonnes et d'autant de lignes que d'arêtes contenant pour chaque arête, l'indice du sommet source et l'indice du sommet cible. Pour les graphes non orientés, on prend comme convention de placer l'indice le plus faible en premier. De plus, les arêtes sont triées en fonction de la valeur du premier indice.

Enfin, il est possible de considérer les faces du graphe, voire les cellules du solide associé. On s'écarte de la définition de graphe pour considérer plutôt des complexes cellulaires euclidiens, mais c'est assez pratique pour manipuler des polyèdres, ou des tessellations (telles que Voronoï).

faces : un tableau de cellules, dans lequel chaque cellule contient les indices des **sommets** d'une face. Si le polyèdre est régulier, toutes les faces ont le même nombre de sommets, et on peut remplacer le tableau de cellules par un tableau numérique.

cells : un tableau de cellules, dans lequel chaque cellule contient les indices des **faces** de chaque volume.

2.2 Structure de données

Certaines fonctions acceptent une structure de données, qui contient les champs suivants (ceux avec une étoile sont optionnels) :

nodes un tableau contenant les coordonnées des sommets

edges un tableau contenant les indices des sommets de départ et de fin de l'arête

***faces** le tableau des indices de sommets pour chaque face

***cells** le tableau des indices de faces pour chaque cellule

3 Création de graphes

Quelques fonctions qui créent un graphe.

3.1 Création simple

3.1.1 graphStructure

Renvoie une structure de données à partir des différents tableaux de données du graphe

3.2 Graphes géométriques

3.2.1 knnGraph

Crée le graphe des k plus proches voisins d'un ensemble de points.

3.2.2 euclideanMST

Crée l'arbre couvrant de poids minimal (utilise la fonction `prim`).

3.2.3 prim

Calcule un arbre couvrant de poids minimal par l'algorithme de Prim.

3.3 Graphes d'une image

Ces fonctions prennent en entrée une image, en général binaire, et renvoient une structure de graphe, qui peut être utilisée pour visualiser ou appliquer d'autres types de traitements sur l'image.

Il existe aussi la fonction `imRAG`, qui calcule le graphe d'adjacence d'une image labellisée, et qui est présente dans le module `imMeasures` de la bibliothèque IMAEL.

3.3.1 imageGraph

Crée le graphe équivalent d'une image

3.3.2 boundaryGraph

Renvoie la frontière d'une image binaire sous la forme d'un graphe.

3.3.3 gcontour2d

Crée un graphe 2D (sommets, arêtes et faces) à partir d'une image 2D binaire. Permet de dessiner le contour des pixels de manière vectorielle.

3.3.4 gcontour3d

Crée un graphe 3D (sommets, arêtes et faces) à partir d'une image 3D binaire. Permet de dessiner le contour des voxels de manière vectorielle.

3.3.5 vectorize

Transforme un squelette binaire en un graphe, en chaînant les pixels adjacents.

4 Manipulation de graphes

4.1 Extraction d'informations

Fonctions pour calculer quelques paramètres descriptifs simples d'un graphe.

4.1.1 getNodeDegree

Renvoie le degré d'un sommet, c'est à dire le nombre d'arêtes connectées à ce sommet. On considère que le graphe n'est pas orienté.

4.1.2 getNodeInnerDegree

Renvoie le degré entrant d'un sommet, c'est à dire le nombre d'arêtes qui arrivent sur ce sommet. On considère que le graphe est orienté.

4.1.3 getNodeOuterDegree

Renvoie le degré sortant d'un sommet, c'est à dire le nombre d'arêtes qui émanent sur ce sommet. On considère que le graphe est orienté.

4.1.4 getNeighbourNodes

Renvoie l'ensemble des sommets adjacents à un sommet donné, dans un graphe non orienté.

4.1.5 getNeighbourEdges

Renvoie l'ensemble des arêtes adjacentes à un sommet donné, dans un graphe non orienté.

4.2 Opérations de bas niveau

Quelques opérations élémentaires de transformations de graphes. Les sommets et les arêtes sont identifiés par leur(s) indice(s) dans les tableaux de données.

4.2.1 addFace

Ajoute une face à un graphe. La face est déterminée par l'ensemble des indices des sommets.

4.2.2 removeNode

Supprime un sommet du graphe, ainsi que l'ensemble des arêtes connectées à ce sommet.

4.2.3 removeNodes

Supprimer plusieurs sommets d'un graphe, ainsi que l'ensemble des arêtes connectées à un ou plusieurs de ces sommets.

4.2.4 **removeEdge**

Supprime une arête dans un graphe.

4.2.5 **removeEdges**

Supprime une arête dans un graphe.

4.2.6 **removeDoubleEdges**

Supprime les arêtes « doubles », qui partagent les mêmes sommets de départ et d'arrivée, pour n'en garder qu'une. Si on a évité d'avoir des boucles (même sommet de départ et d'arrivée), on obtient un graphe simple.

4.3 Opérations générales

Plusieurs fonctions de fusion ou de simplification de graphes, qui ne font intervenir que la topologie du graphe (on n'utilise pas les coordonnées des points).

4.3.1 **mergeGraphs**

Fusionne deux graphes, en raboutant les différents tableaux de données, et en convertissant les indices des tableaux des arêtes et des faces.

4.3.2 **mergeNodes**

Fusionne plusieurs sommets dans un graphe, et convertit les références des arêtes.

4.3.3 **simplifyGraph**

Simplifie un graphe en supprimant les sommets multiples.

4.3.4 **simplifyLines**

Supprime les sommets de degré 2 dans un graphe. Utilisation : dans un graphe obtenu après une ligne de partage des eaux, remplace les frontières des cellules par des arêtes joignant les points triples.

4.3.5 **minGraph**

La même chose que `simplifyLines`, une des deux fonctions doit être de trop...

5 Opérations particulières

5.1 Filtrage sur les sommets d'un graphe

Ces fonctions travaillent sur des graphes valués. Pour chaque sommet, elles récupèrent les valeurs du sommet et de ses voisins, et appliquent un traitement mathématique.

5.1.1 grMean

Calcule la moyenne sur les étiquettes des voisins.

5.1.2 grMedian

Calcule la médiane sur les étiquettes des voisins.

5.1.3 grDilate

Calcule une dilatation morphologique sur les étiquettes des voisins. Équivalent à calculer la valeur max.

5.1.4 grErode

Calcule une érosion morphologique sur les étiquettes des voisins. Équivalent à calculer la valeur min.

5.1.5 grOpen

Calcule une ouverture morphologique sur les étiquettes des voisins.

5.1.6 grClose

Calcule une fermeture morphologique sur les étiquettes des voisins.

5.2 Opérations sur les graphes géométriques

Ces opérations utilisent le fait que les sommets sont des points 2D ou 3D.

5.2.1 mergeNodesMedian

Fusionne plusieurs sommets dans un graphe, en utilisant comme sommet résultat un sommet dont les coordonnées sont les médianes des coordonnées des sommets d'entrée. Convertit les références des arêtes.

5.2.2 removeMultiplePoints

Supprime des configurations particulières en triangle dans un graphe.

5.2.3 cleanGraph

Retourne un graphe « nettoyé », c'est-à-dire sans points triples, sans arêtes doubles, et sans sommet de degré 2.

5.2.4 clipGraph

Calcule l'intersection du graphe géométrique

5.2.5 getGraphFace

Renvoie une face dans un graphe sous la forme d'un polygone 2D ou 3D.

5.3 Diagrammes de Voronoï

Quelques fonctions d'adaptation pour manipuler des résultats de calcul de diagramme de Voronoï sous forme de graphe.

5.3.1 boundedVoronoi2d

Calcule un diagramme de Voronoï dans le plan et renvoie le résultat sous la forme d'un graphe géométrique 2D. Le diagramme est borné par une boîte (coordonnées min et max dans chaque direction), ce qui permet d'avoir un résultat contenant uniquement des faces bornées.

5.3.2 centroidalVoronoi2d

Crée un diagramme de Voronoï centroïdal dans le plan.

5.3.3 voronoi2d

Calcule un diagramme de Voronoï dans le plan et renvoie le résultat sous la forme d'un graphe géométrique 2D.

6 Fonctions d'affichage

6.1 Affichage général

6.1.1 drawGraph

Dessine un graphe dans la fenêtre courante.

6.1.2 drawGraphEdges

Dessine les arêtes d'un graphe dans la fenêtre courante.

6.1.3 drawGraphFaces

Dessine les faces d'un graphe dans la fenêtre courante.

6.1.4 drawDigraph

Dessine un graphe bipartite, donné comme deux listes de sommets, et une liste d'arêtes qui associent un sommet de la liste de départ à un sommet de la liste d'arrivée.

6.1.5 drawDirectedEdges

Dessine des arêtes orientées (avec une flèche au milieu) dans la fenêtre courante.

6.2 Affichage d'informations sur le graphe

6.2.1 drawEdgeLabels

Affiche les étiquettes associées à chaque arête.

6.2.2 drawNodeLabels

Affiche les étiquettes associées à chaque sommet.

6.2.3 graph2Contours

Convertit un graphe en un ensemble de contours.

6.2.4 drawSquareMesh

Dessine un graphe avec uniquement des faces carrées.

6.2.5 patchGraph

Transforme un graphe 3D en patch Matlab.