8INF914 Analyse visuelle

**Rapport de projet final**

Approche orientée pixel

CHABALIER Nicolas  
COCCHI Elian  
DUCHENE Nicolas

Introduction

L’approche orientée pixel en analyse visuelle est une méthode cherchant à optimiser l’espace occupé par la représentation d’un graphe tout en gardant une représentation cohérente des données pour qu’elles soient faciles à analyser à l’œil nu.

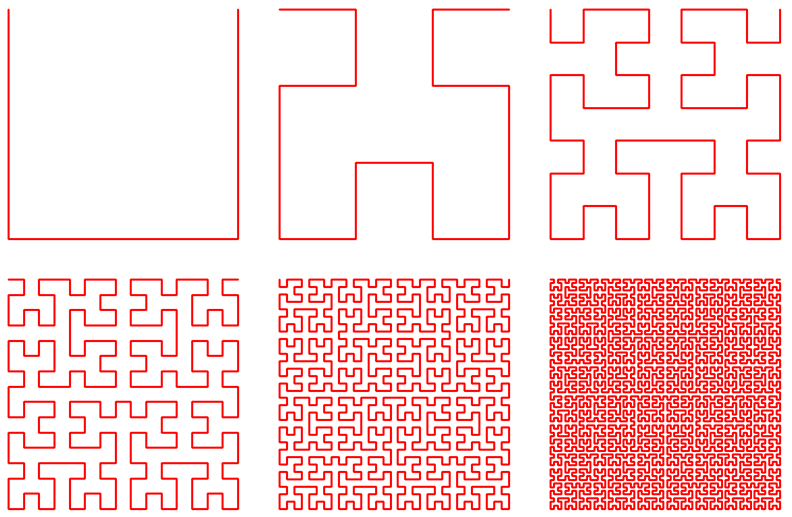
Le principe est que chaque donnée est représentée par un espace à l’écran considéré comme unitaire (souvent un simple carré ou disque, d’où le terme de représentation ‘pixel’). Il s’agit ensuite de déterminer un schéma le long duquel les pixels représentant les données seront placés. Le choix de ce schéma est très important, car c’est lui qui fait tout l’intérêt de la représentation : un bon schéma est compacte et lisible.

Dans la suite, nous verrons donc plusieurs de ces représentations, que nous avons implémentées avec tulip sous forme de plugins à utiliser sur des jeux de données importés.

Courbes fractales

L’utilisation de courbes fractales pour représenter les jeux de données est très intéressante. En effet, elles permettent une très bonne compacité (certaines courbes fractales ont une dimension proche de 2) et permettent d’avoir une logique dans la représentation des données : Il suffit de suivre la courbe du début à la fin pour voir toutes les données dans l’ordre dans lequel elles ont été placées. D’autre part, elles permettent d’avoir une taille toujours adaptée à celle du jeu de données : on détermine le nombre d’itérations de la fractale ont doit faire pour représenter l’ensemble du jeu de données, sans aller plus loin.

Le principe est simple : on choisit un type de courbe fractale (nous avons ici utilisé les courbes de Hilbert), puis on la parcourt en plaçant un sommet représentatnt une donnée à chaque changement de direction.

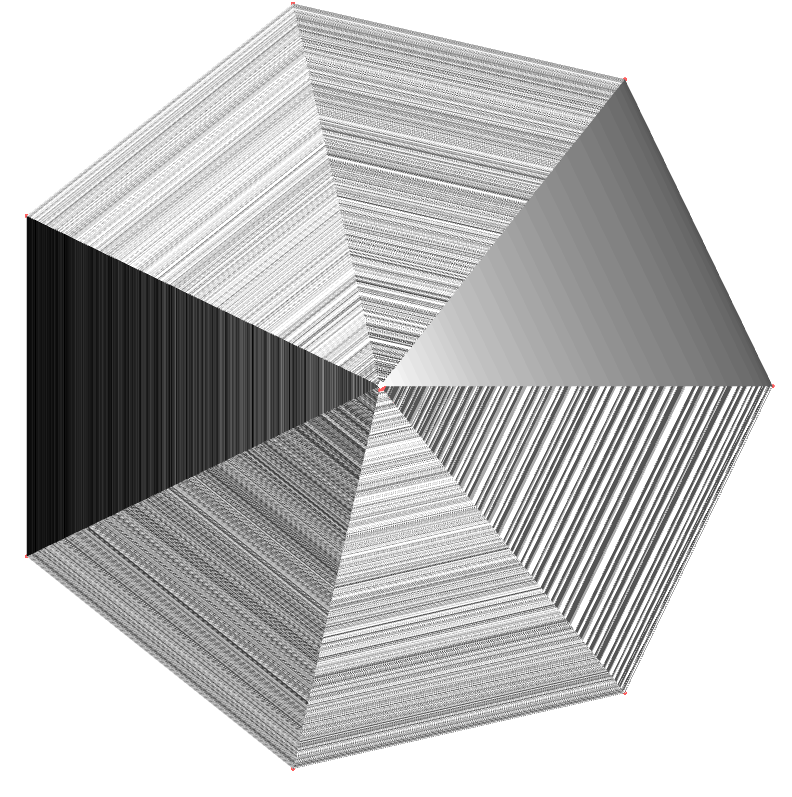


*Courbe fractale de* Hilbert *aux itérations 1 à 6*

Notre plugin pour cette approche s’appelle *HilbertPlugin.py*, pour l’utiliser, il suffit de lancer le script python dans tulip pour créer le plugin, puis d’utiliser le plugin avec l’interface graphique. Il suffit alors de choisir la variable en fonction de laquelle on veut que les données soient triées, puis de lancer le plugin pour qu’il crée une vue orientée pixel en utilisant une courbe de Hilbert, calculée à l’itération minimum nécessaire à la représentation de toutes les données.

Courbe circulaire et datatube

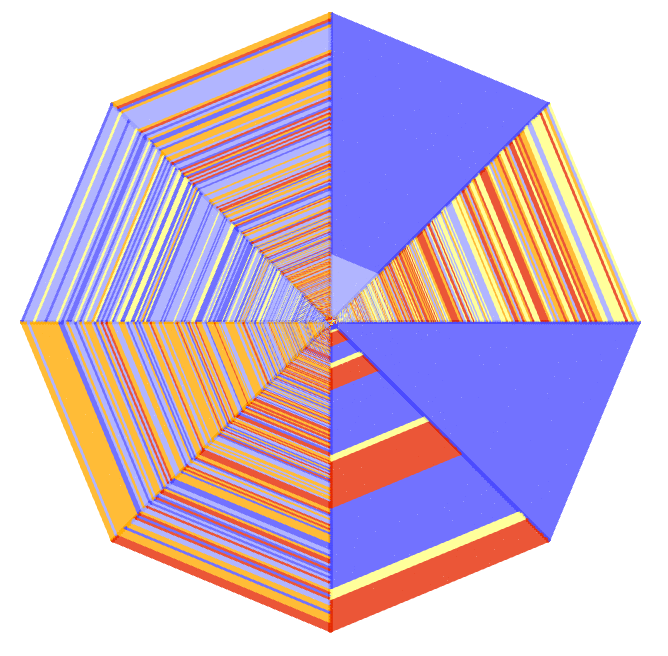
Une deuxième approche que nous avons implémentée est celle de la courbe circulaire. Ici, le but est de représenter les données en forme de cercle autour d’un point centrale (en fait plutôt des polygones réguliers à autant d’arêtes que de variables représentant les données). Chaque variable est représentée le long d’un « quartier » du cercle ; une donnée est donc entièrement représentée par une ligne faisant le tour complet du cercle. Les valeurs des variables se lisent à l’aide d’un gradient de couleur.



*Exemple de représentation par courbe circulaire d’un jeu de données à 7 variables, en nuances de gris*

De nouveau, cette vue est compacte (ici sous la forme d’un disque plutôt qu’un carré) permettant une lecture rapide. En triant les données en fonction d’une certaine variable (on s’assure alors que les données avec les valeurs extrêmes pour cette variable se trouvent au centre et sur le cercle extérieur), on peut rapidement trouver les variables corrélées et voyant lesquelles ont un dégradé de couleur similaire.

Une évolution de l’approche par courbe circulaire est le datatube. Le principe est le même, cependant plutôt que de simples lignes, les variables sont représentées par trapèzes dont l’épaisseur dépend de sa distance au centre. Ceci à pour but de donner une impression de profondeur au dessin : on dirait un tunnel où les données serait représentées sur les bords.



*Exemple de jeu de données représenté sous la forme d’un datatube*

On voit par exemple dans la figure ci-dessus que les données sur les bords sont plus larges que celles au centre, donnant une impression de profondeur.

C’est la version datatube que nous avons décidé d’implémenter sous forme de plugin tulip. De même que pour la courbe de Hilbert, il faut lancer le fichier *dataTubePlugin.py* pour créer le plugin, qu’on utilise ensuite à l’aide de l’interface graphique. On choisit trois paramètres : la largeur de base des trapèzes (attention, une erreur de tulip empêche d’utiliser des valeurs trop grandes ici, renvoyant simplement un graphe entièrement blanc), la variables selon laquelle trier les données et enfin la plage de couleur pour le coloriage.

Le script pour la vue circulaire simple est fourni aussi, mais pas sous la forme de plugin.