

RDF - TP2 - Contours d'une forme

Introduction

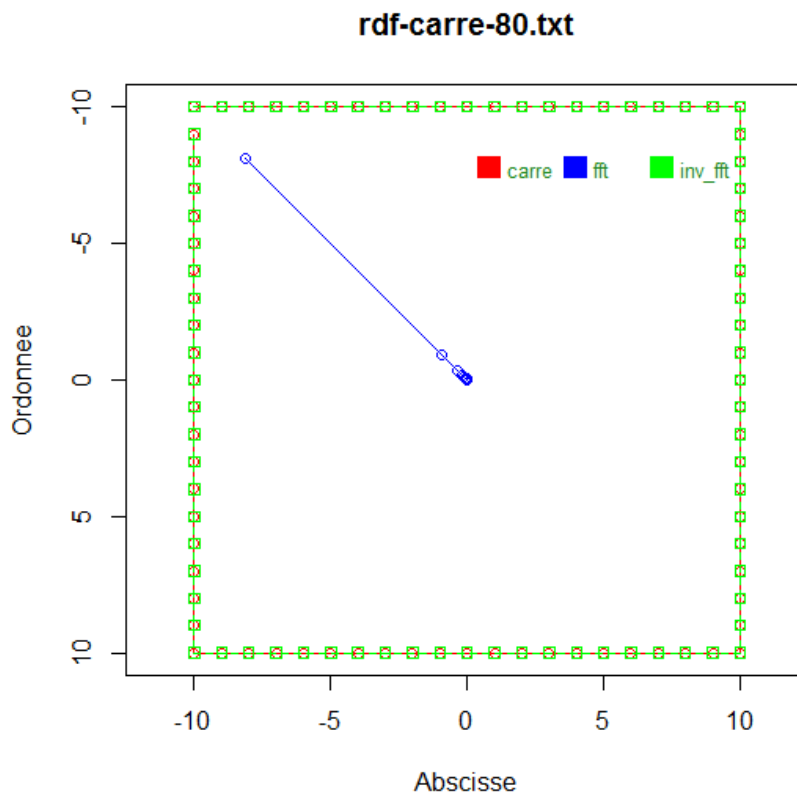
Dans ce second TP, nous allons voir les descripteurs de Fourier et leur impact sur la reconnaissance de formes simples comme un carré, un rectangle, un cercle, ou des formes légèrement plus complexes comme une croix ou le fameux « Patatoïde ».

Descripteurs de Fourier

Le dernier argument nous permet d'obtenir un intervalle défini par les min, max des nombres imaginaires. Cet intervalle définit les limites des coordonnées y.

Les descripteurs de Fourier nous permettent de compacter les données.

On peut voir sur l'image ci-dessous la description de Fourier (courbe bleue) et l'inverse de cette description de Fourier (courbe verte). La description de Fourier représente le carré sous forme compacte et en testant l'inverse de cette description de Fourier on retrouve bien le carré original (superposition du carré vert et le carré rouge (carré d'origine)).

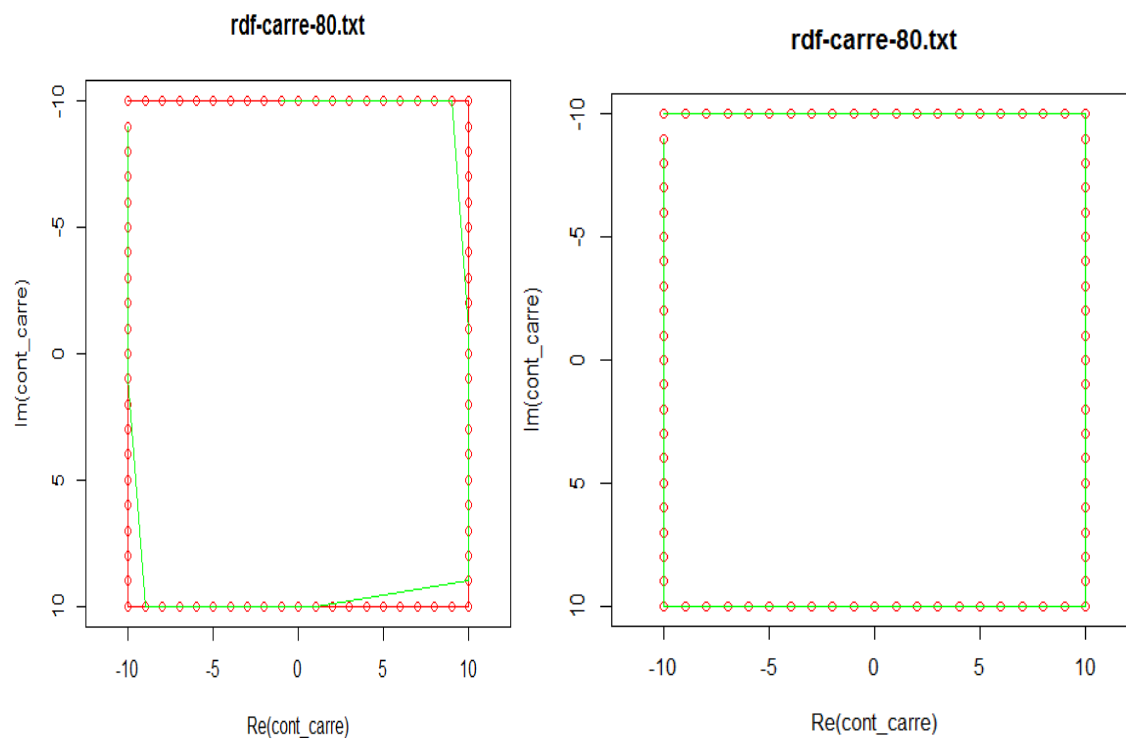


Le carré rouge et vert sont superposés, et la ligne bleue représente Fourier.

Le descripteur de Fourier Z_0 correspond au barycentre de la forme concernée. C'est le premier élément de notre tableau. Il a pour valeur $0+0i$ car il est centré à l'origine.

Grâce à la fonction *rdfAnnuleDescFourier*, on réduit le nombre de points. On peut constater que même avec un ratio de 0.1 (image de gauche) le carré de base est représenté, avec quelques erreurs. L'image de droite avec un ratio de 0.6 respecte parfaitement le carré original.

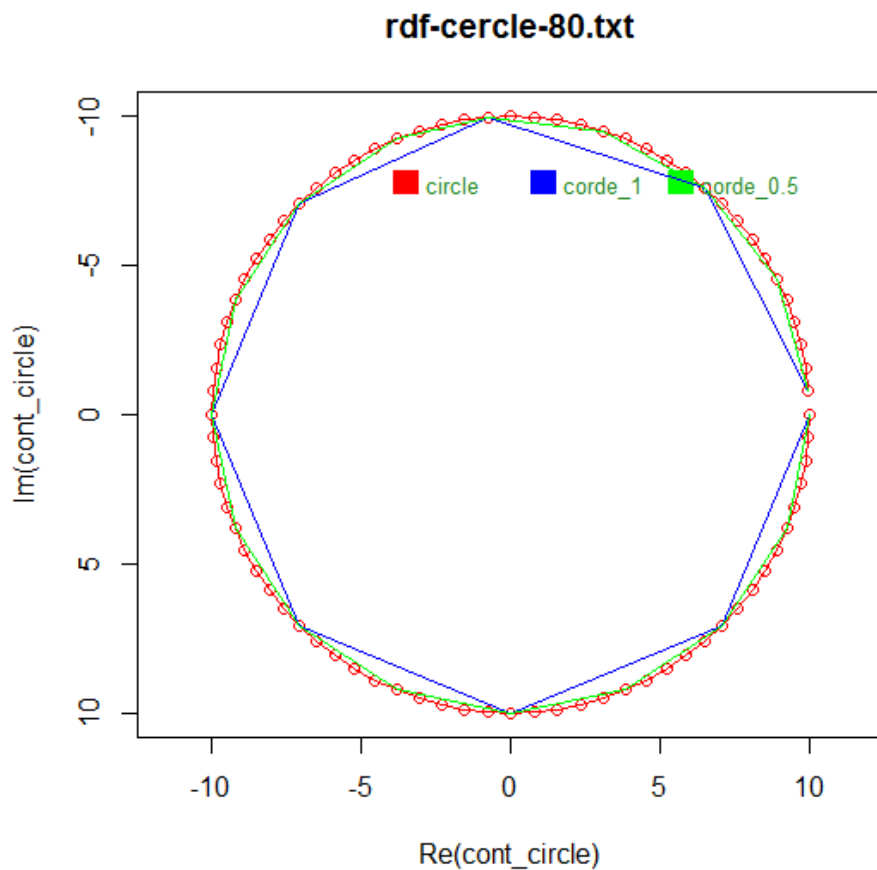
Pour rappel, un ratio de 0.1 élimine 9 descripteurs sur 10, tandis qu'un ratio de 0.6 en élimine seulement 40 %.



Le ratio de 0.1 à gauche, en vert ... Et le ratio de 0.6 à droite, toujours en vert.

Réduction d'une chaîne de contour

Nous avons effectué des tests concernant l'algorithme des cordes sur le contour du cercle (fonction *rdfAlgorithmeCorde*). Sur l'image ci-dessous, nous pouvons constater les résultats de cet algorithme pour la version $dmax=0.5$ en vert et la version $dmax=1$ en bleu.

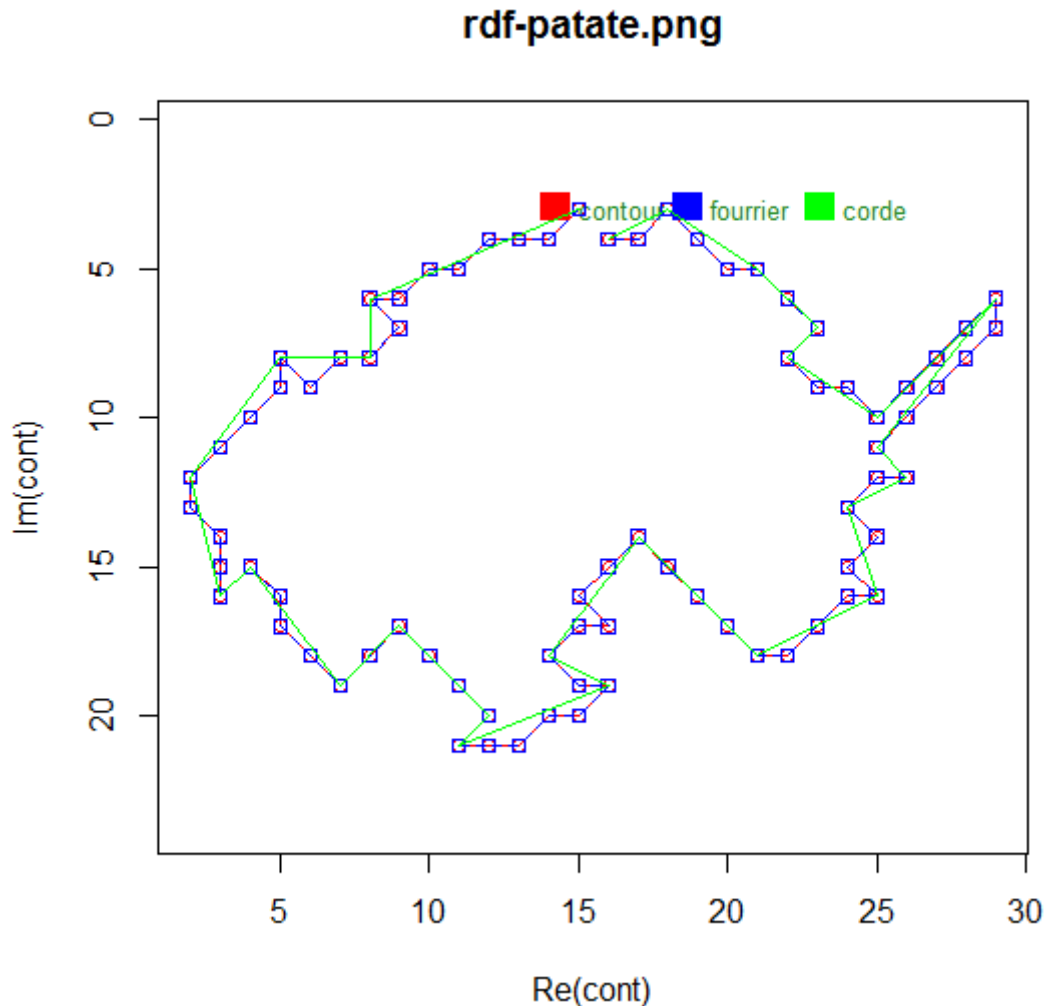


La représentation de l'algorithme des cordes pour un cercle.

On peut observer que petit à petit, on se rapproche de la forme du cercle, de manière très précise. Nous pouvons en conclure que l'algorithme des cordes est très efficace quand il s'agit de déterminer une forme circulaire.

Comparaison des deux approches

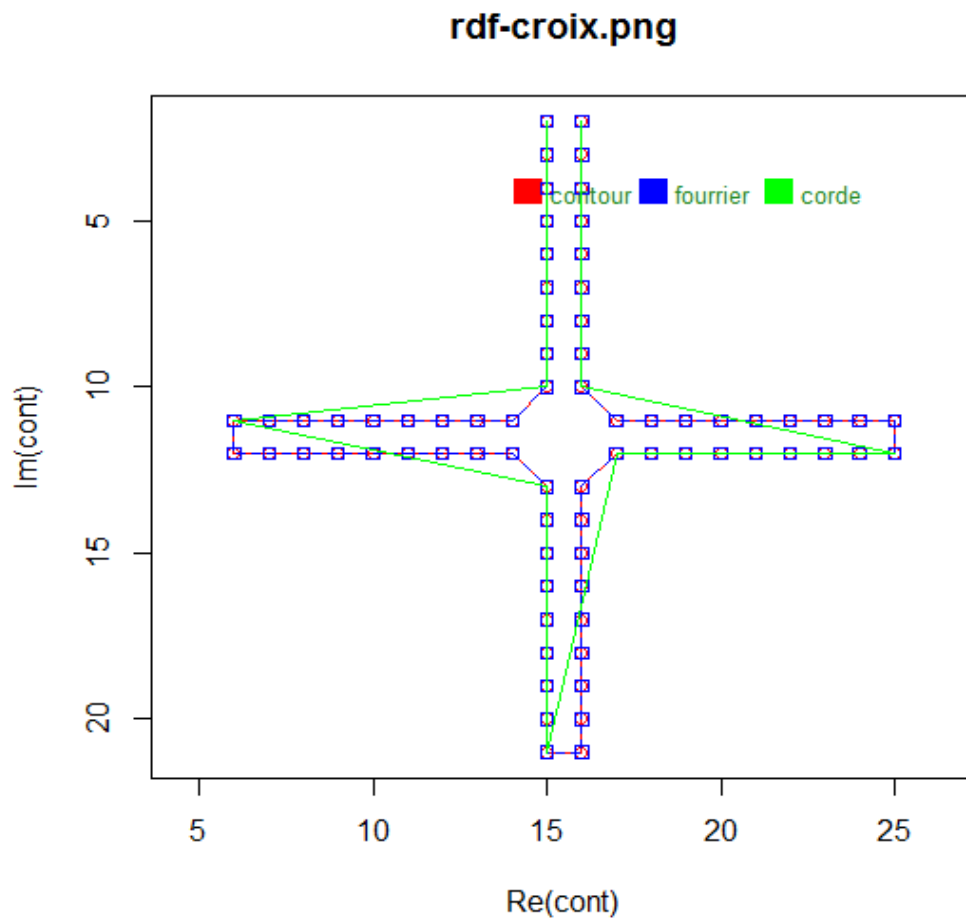
Image *patate.png* :



Sur cette image sont représentés **le contour de la forme** présente sur l'image (en rouge), le résultat de la détection de contour par **Fourier** (en bleu) et le résultat de la détection de contour par **l'algorithme de la corde** (en vert).

Par ces courbes, on peut observer que le contour est plus régulier et précis avec **Fourier**. En effet, l'algorithme des cordes se contente de calculer un segment avec un ensemble de points plus ou moins proches pour avoir un contour avec les moins de points possible. Et donc, cette perte de points quand ils n'ont pas une trajectoire régulière fait que le contour n'est pas précis.

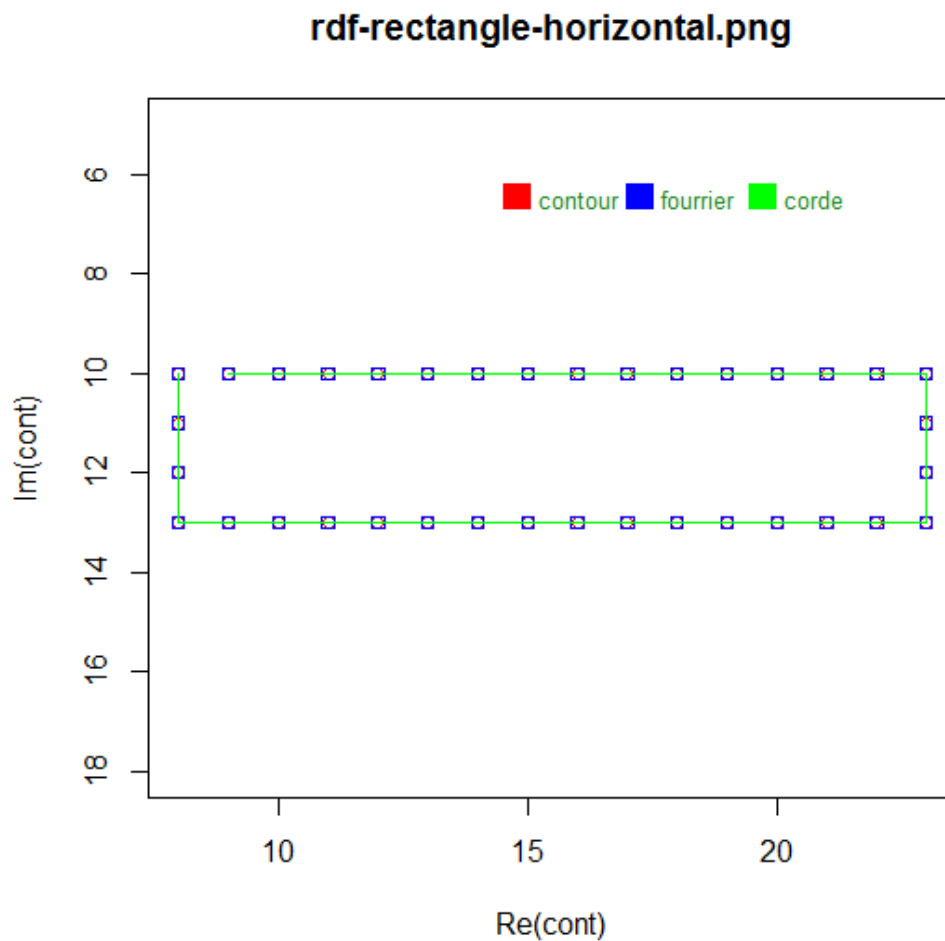
Image *croix.png* :



Les deux algorithmes, employés pour déterminer la forme « Croix.jpg »

Pour cette forme, on remarque que **Fourier** est également plus efficace que l'algorithme des cordes. En effet, l'algorithme des cordes relie entre deux un certain nombres de pointes qui ne sont pas forcément reliés entre eux dans la forme initiale, ce qui provoque cette forme de croix *pointue*.

Image *rectangle-horizontal.png* :

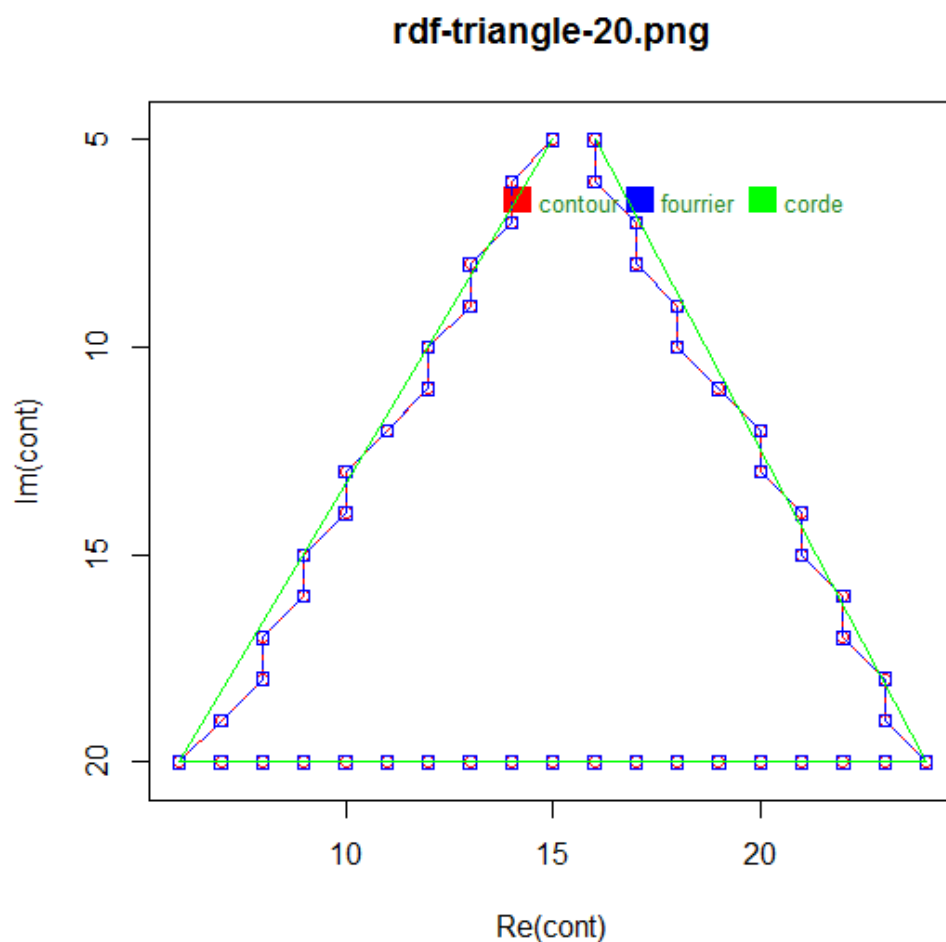


Les deux algorithmes, employés pour déterminer la forme « rectangle-horizontal.jpg »

Pour cette forme, on remarque que les 2 méthodes sont équivalentes en raison de la forme régulière du contour. Pour déterminer le contour du point de vue de **l'algorithme des cordes**, il lui suffit de garder uniquement 4 points qui sont les 4 coins du rectangle. Il serait donc **plus avantageux** d'utiliser cette approche, car moins gourmande en performances.

Dhersin Jérôme
Badache Yassine

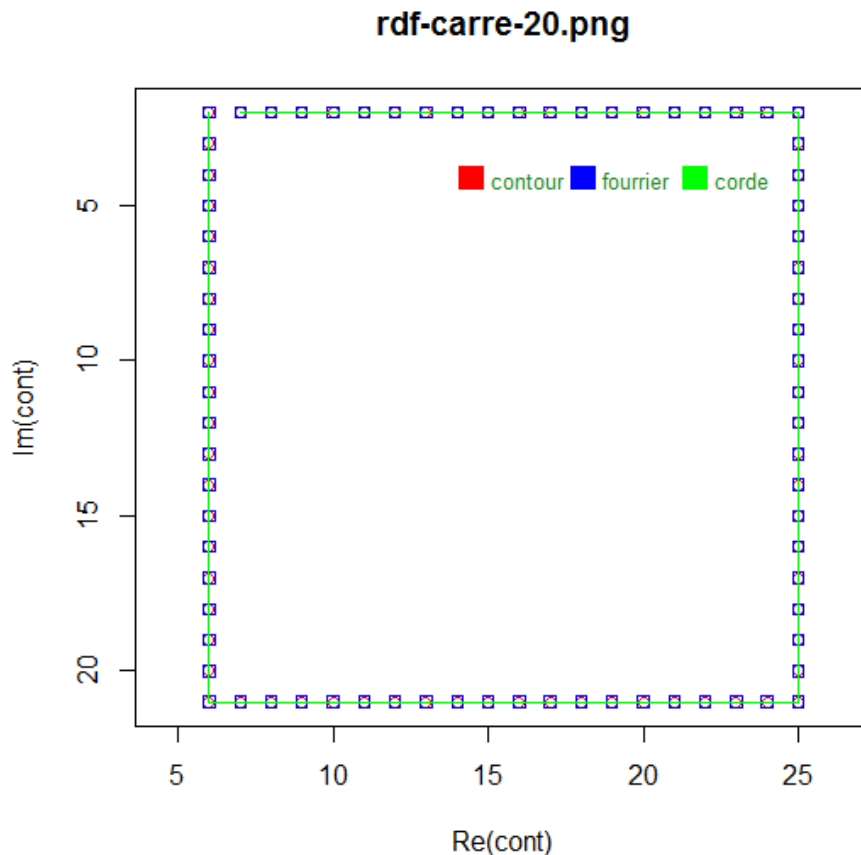
Image *triangle-20.png* :



Les deux algorithmes, employés pour déterminer la forme « triangle-20.jpg »

Pour cette forme, on remarque que **Fourier** est, là encore, plus efficace, car on peut voir que les côtés du triangle ne sont pas réguliers. La conséquence en est que l'algorithme des cordes supprime une partie des points sur les côtés orientés, de un sur deux à un sur trois dans certaines circonstances.

Image *carre-20.png* :



Les deux algorithmes, employés pour déterminer la forme « carre-20.jpg »

Pour cette image on obtient le même résultat que pour l'image du rectangle et donc les 2 approches se valent, malgré l'efficacité de **l'algorithme des cordes** avec uniquement les 4 coins à considérer.

Conclusion

Ces travaux pratiques ont su nous faire comprendre l'importance des descripteurs de Fourier et du compromis qu'il y a à faire entre la précision d'une reconnaissance (*nombre de descripteurs à conserver*) et la performance du calcul.

De même, il nous a permis d'analyser le comportement de cet algorithme en comparaison avec l'algorithme des cordes, ce qui nous a permis de déduire que bien qu'efficace, l'algorithme des cordes était davantage utile lorsqu'il s'agit de déterminer des formes *régulières* comme des carrés ou des cercles, alors que les descripteurs de Fourier se prêtent à plus ou moins tout type d'exercice, sous réserve d'un nombre de descripteurs suffisants.