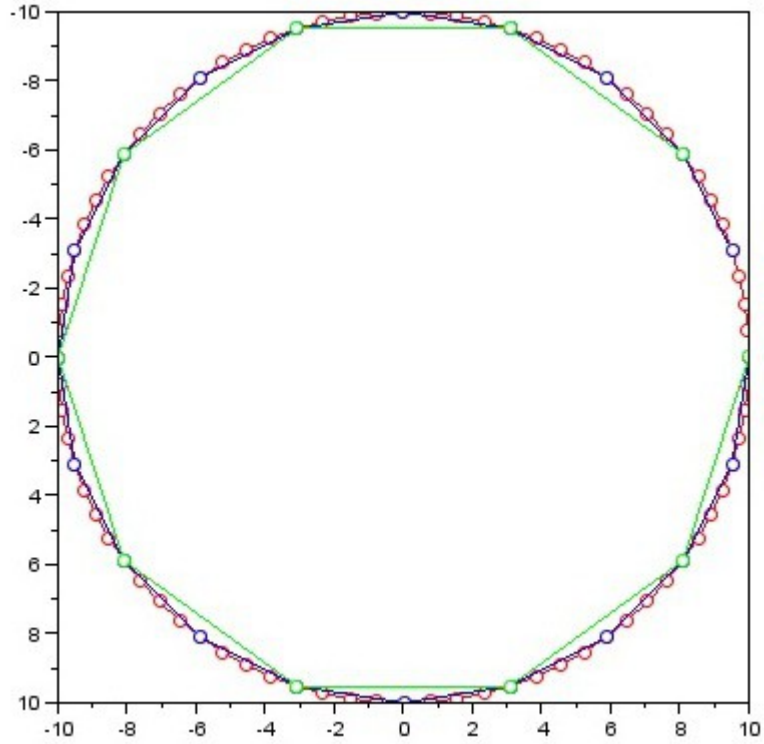
RDF TP2 : codage d'un contour

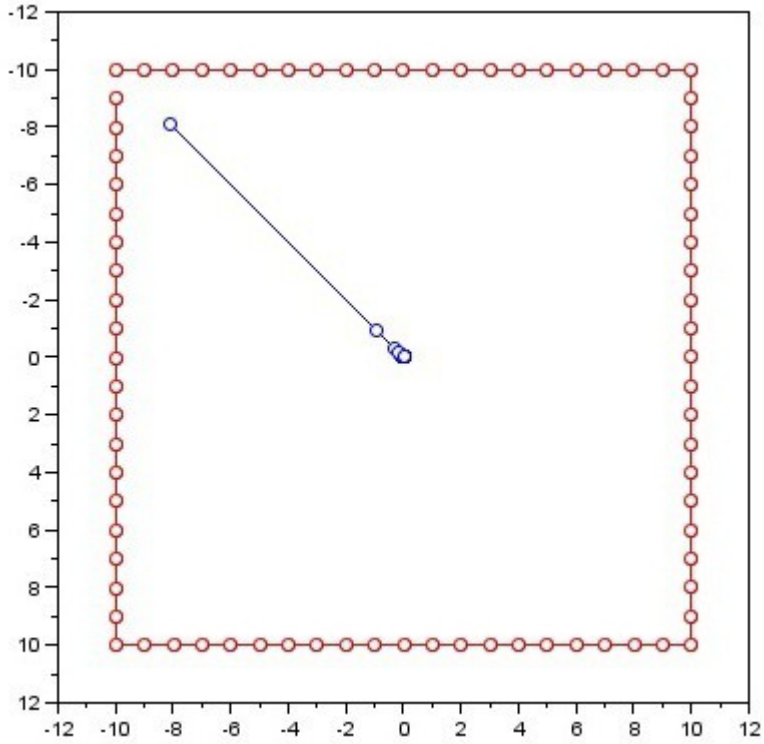
binôme : Benjamin Ruytoor et Aurore Allart
date : 26 janvier 2013

1. Descripteurs de Fourier

Voici la représentation d'un cercle en prenant en compte que certains point du contour : avec tous les points (rouge), avec un point sur 4 (bleu) et enfin avec un point sur 8 (vert) :

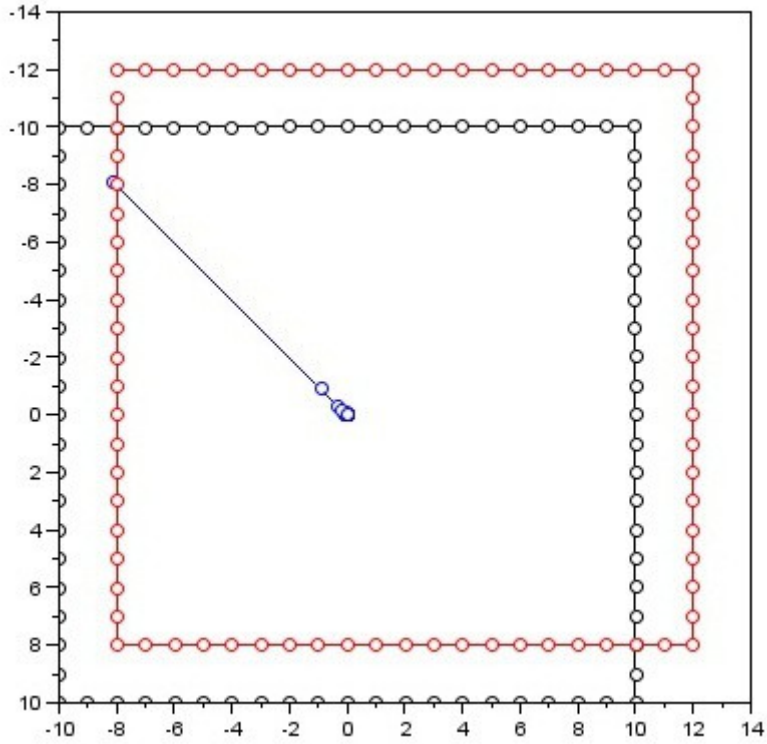


Les descripteurs de Fourier permettent de coder la signature d'un contour en une fonction périodique. La fonction élimine parfois un point de la liste, quand celle-ci est de taille impair. Voici les descripteurs pour la figure d'un carré :



Le barycentre Z0 est à l'indice 1 dans la matrice résultante de la fonction rdfDescFourier. La fonction rdfValeurDescFourier permet de retourner la valeur dans la matrice à l'indice demandée. Par exemple, si on demande de retourner dans les descripteurs de Fourier la valeur de l'indice 0 (lien avec l'indice de Z), il retournera le barycentre.

Si on modifie la valeur de Z0 dans la matrice de descripteurs avant l'inversion, cela aura pour effet de translater la forme. En voici un exemple :

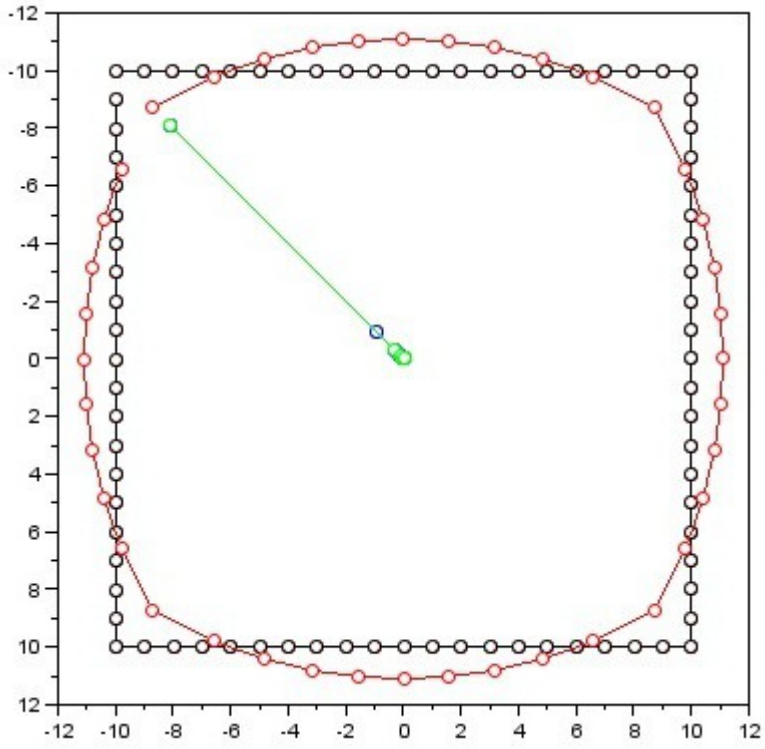


Le carré noir étant le carré initial, et le rouge celui de l'inversion. On voit nettement l'influence du barycentre.

2. Filtrage des descripteurs de Fourier

Pour annuler certains descripteurs de Fourier voici la fonction et un exemple d'utilisation sur le carré.

```
// annule certains des descriteurs de Fourier
function annule=rdfAnnuleDescFourier(desc, ratio)
    annule = desc;
    if ratio == 0 then
        annule = zeros(size(desc,1),1);
    else
        nbValSup = size(desc,1)- size(desc,1)*ratio;
        annule = desc(1:nbValSup);
    end
endfunction
```

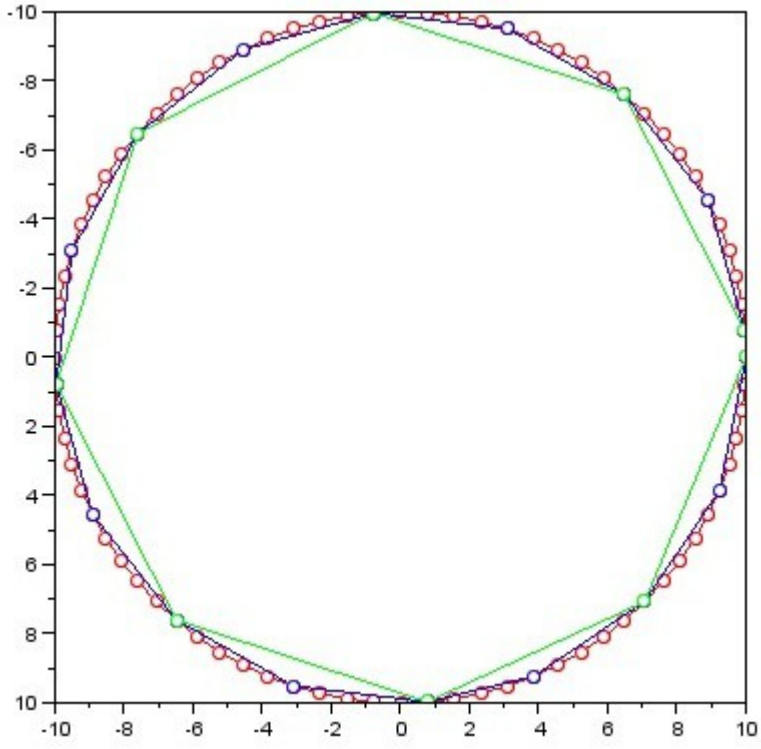


3. Réduction d'une chaîne de contour

Voici le bout de code pour l'implémentation de l'algorithme de corde :

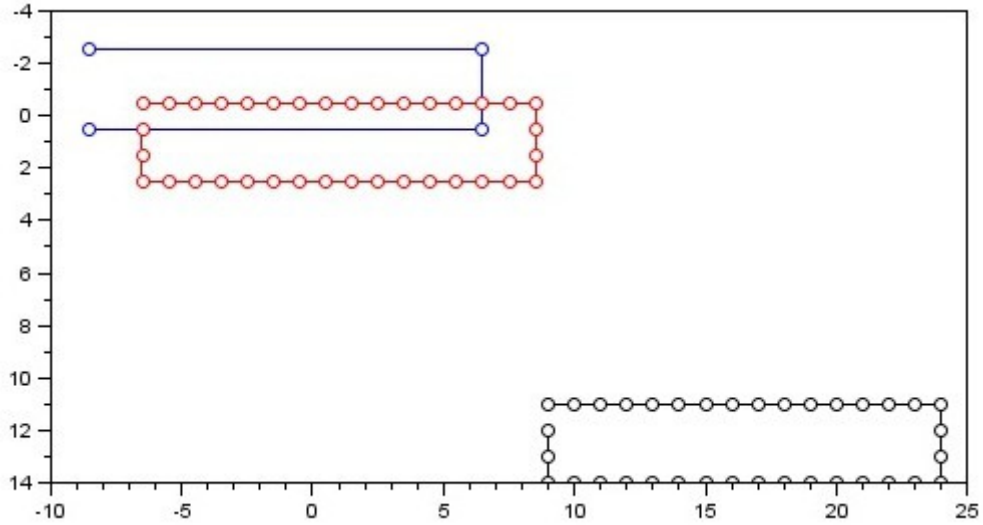
```
if (real(debut)-real(fin))==0 then
    d = abs(real(cont)-real(debut));
else
    if (imag(debut)-imag(fin))==0 then
        d = abs(imag(cont)-imag(debut));
    else
        a = (imag(debut)-imag(fin))/(real(debut)-real(fin));
        b = imag(fin) - a*real(fin);
        d = (abs(real(cont)*a - imag(cont) +b))/(sqrt(1 + a*a));
    end
end
```

Ainsi que la représentation graphique avec le contour initial sans changement (rouge), le contour avec une distance de 1 pixel (vert) et enfin celui avec une distance de 0,5 pixel (bleu).

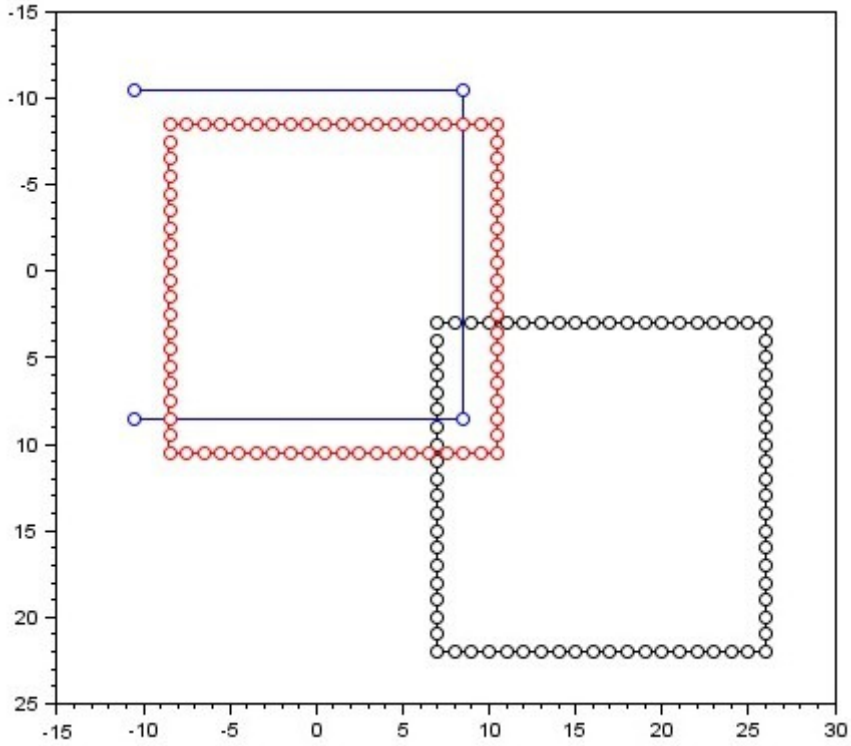


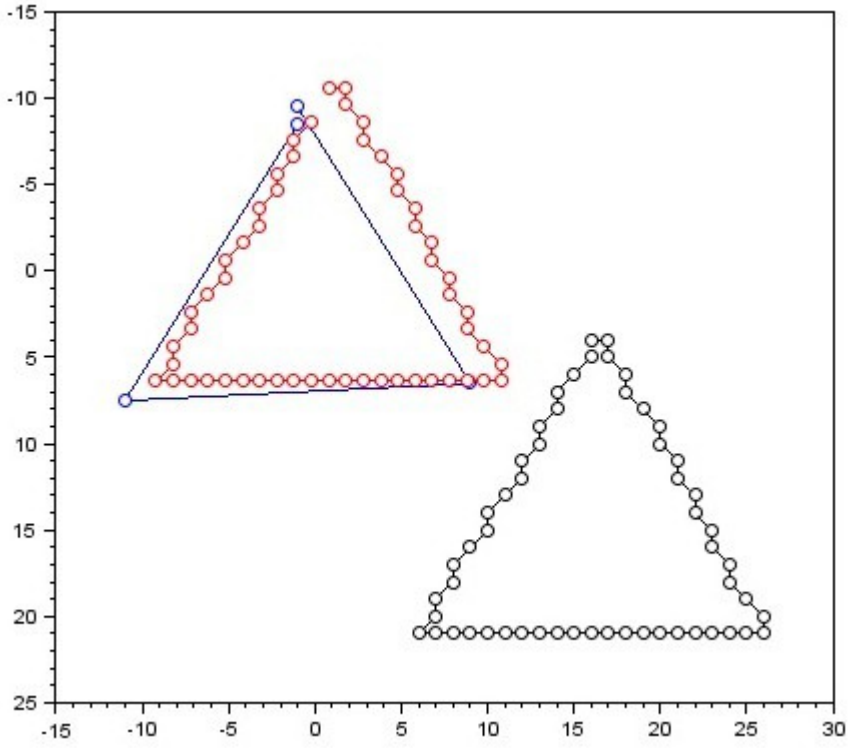
4. Comparaison des deux approches

Pour le rectangle, nous avons utilisé les paramètres optimums pour les descripteurs de Fourier et pour l'algorithme de la corde, cela ne charge rien si on prend une dmax = 0,5 ou 1, les points choisis sont identiques, car le rectangle est formée de droite. Voici la représentation graphique des 2 approches avec un pixel décalé pour une meilleure lisibilité. Le contour noir représente le contour initial, le rouge celui des descripteurs de Fourier et enfin le bleu celui avec l'algorithme de la corde.

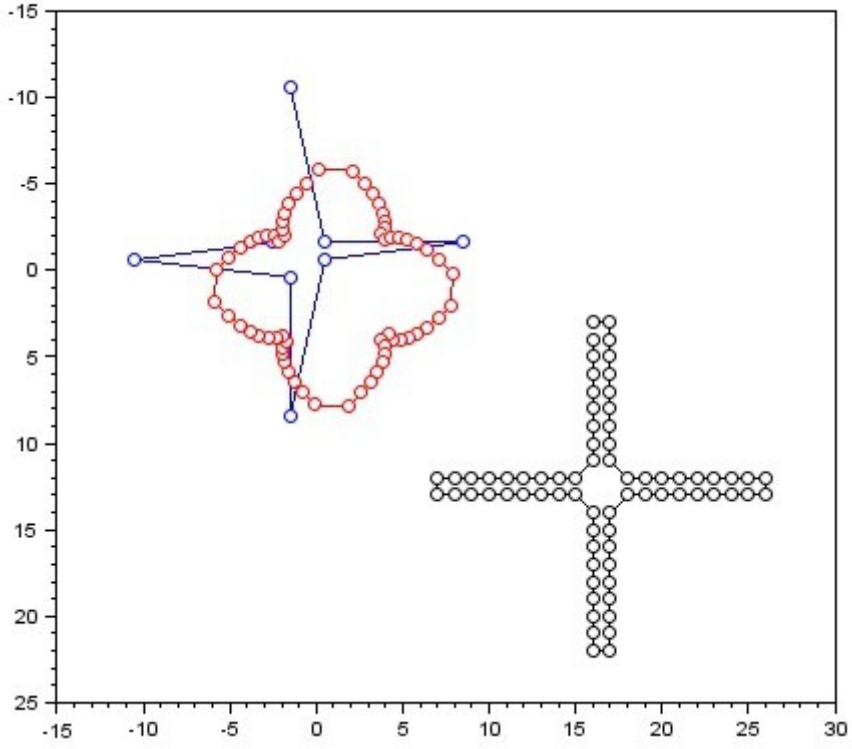


Pour le carré et le triangle, c'est pareil que le rectangle.

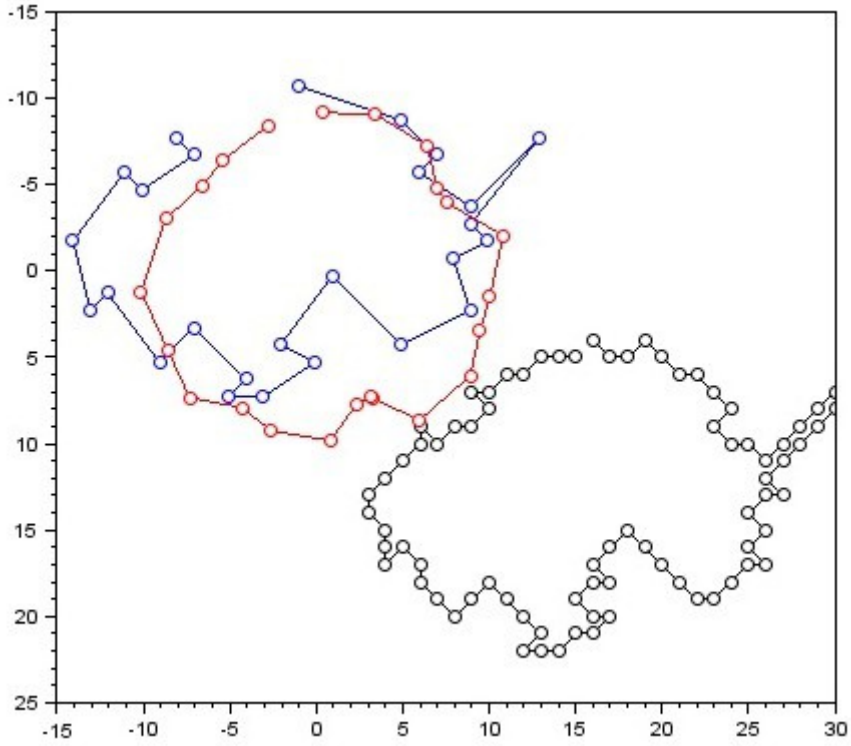




Par contre pour la croix, nous avons pu réduire le nombre de descripteur de Fourier tout en visualisant toujours une croix, certes non identique à la première mais en si approchant.



Pour la patate, avec l'algorithme de la corde, on voit très bien que la forme ne ressemble pas à grand chose, cependant avec les descripteurs de Fourier, on s'aperçoit quand même que la forme est proche d'un cercle.



Conclusion :

Les 2 démarches pour approcher un contour d'une forme sont intéressantes. L'algorithme de la corde est très utile pour des formes rectilignes comme le triangle ou la croix, car elle permet de réduire le nombre de point, tout en gardant la forme de la figure. Cependant pour les formes comme la patate, cet algorithme prend en compte des points extrêmes et non représentatives de la figure (la pointe en bleu sur la figure ci-dessus). Par contre la méthode des descripteurs de Fourier est meilleure pour ces formes car elle supprime justement ces points et est médiocres avec des formes rectilignes.