VISI 201
Rey Lukas
Tuteur : M. Lachaud

Géométrie discrète, combinatoire des mots
Comment trouver le nombre de côté d'un polygone dans une image ?

Introduction :

Nous considérerons une image d'un polygone constitué de pixels noirs sur un fond blanc, en supposant que les bords du polygones ne touchent pas les bords de l'image.
L'objectif est d'extraire le contour du polygone puis de l'analyser afin d'y trouver des droites que nous pourrons compter pour obtenir le nombre de côtés du polygone.

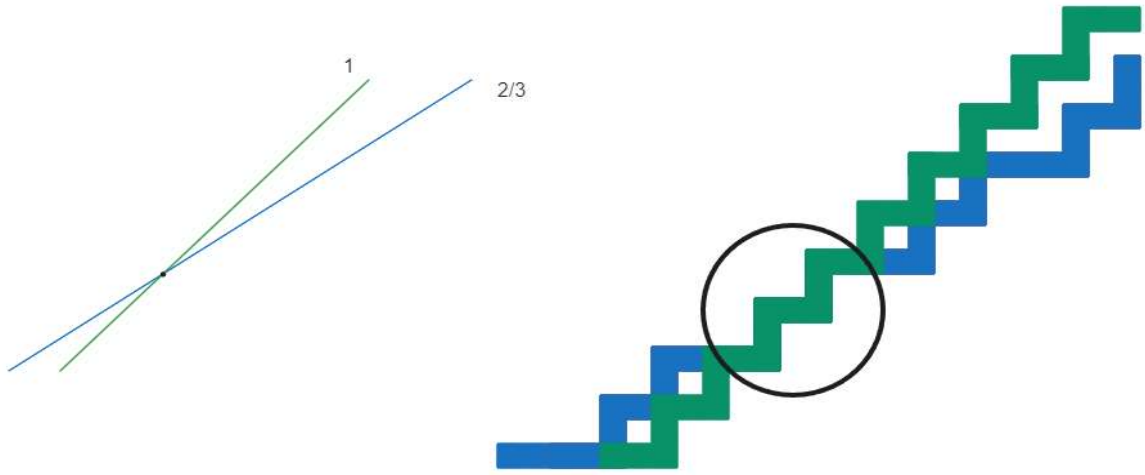
Sommaire :

-Géométrie avec des pixels
-Les mots de Christoffel
-Reconnaissance de droites
-Droites dans un mot de contour
-Obtenir un mot de contour à partir d'une image
-Conclusion

Géométrie avec des pixels

1) Géométrie difficile

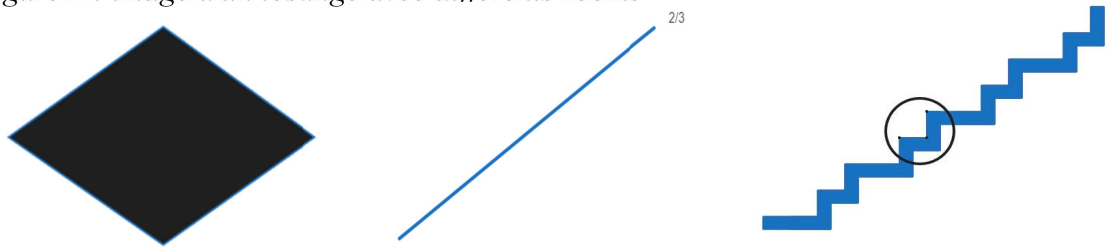
Figure 1 : droites de pente 1 (vert) et de pente 2/3 (bleu) qui s'intersecte.



Certains résultats de la géométrie euclidienne ne tiennent plus lorsque l'on utilise des pixels :
par exemple la figure 1 montre que deux droites non parallèles ne se coupent pas forcément en un seul point lorsqu'elles sont représentées par des pixels.

2) Problème d'échelle.

Figure 2 : image d'un losange avec différents zooms

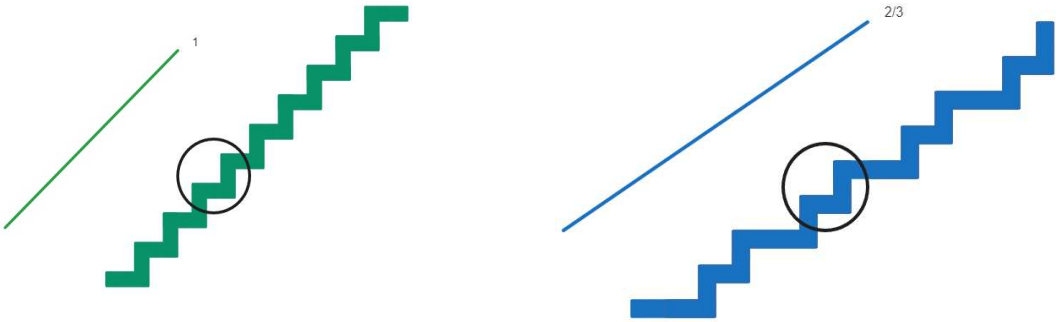


[(0, 0), (1, 0), (2, 0), (2, 1), (3, 1), (3, 2), (4, 2), (5, 2), (5, 3), (6, 3), (6, 4), (7, 4), (8, 4), (8, 5), (9, 5), (9, 6), (10, 6), (11, 6), (11, 7), (12, 7), (12, 8)]

L'humain possède une vision globale des choses ce qui nous permet d'identifier aisément le nombre de côtés d'un polygone. Or un ordinateur ne possède pas cette vision, il ne peut voir qu'une suite de points, comme la figure 2 le montre en détaillant une image.
Nous devrons donc trouver un moyen de retrouver une vision plus large en utilisant des algorithmes.

3) Problème de précision

Figure 4 : Deux droites différentes qui peuvent sembler identiques si nous considérons uniquement les points compris dans les cercles noir.



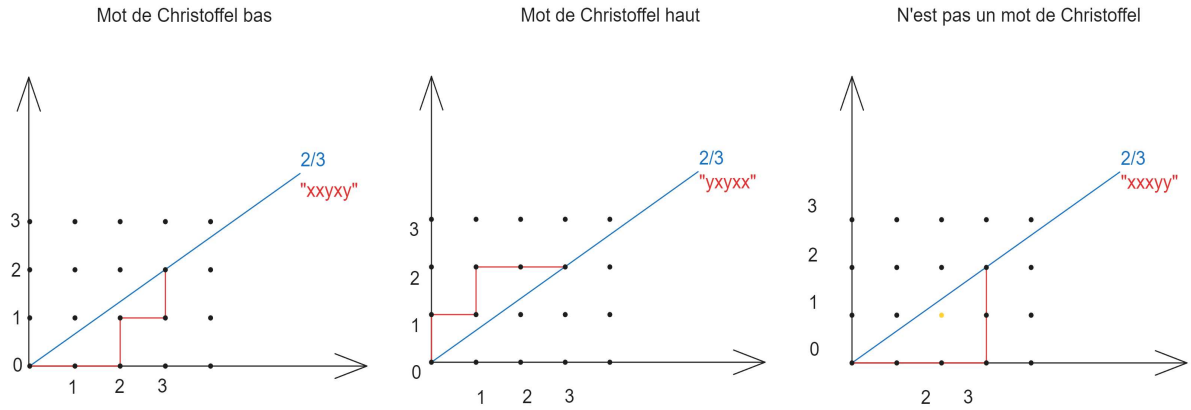
Nous allons aussi devoir manipuler suffisamment de données pour pouvoir avoir une analyse précise.
La figure 4 montre le problème que pourrait engendrer un nombre insuffisant de pixels.

Les mots de Christoffel

1) définition subjective

Manière d'approcher une droite avec des coordonnées entières

Figure 5 :



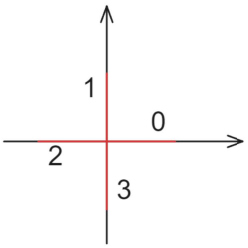
Le chemin représenté par un mot de Christoffel essaie toujours de se rapprocher au maximum de la droite en passant par des coordonnées entières.
Un pas horizontal vers la droite est représenté par un « x » et un pas vertical vers le haut par un « y ».
La dernière image de la figure 5 ne correspond pas à un chemin d'un mot de Christoffel car le point en jaune est une coordonnée entière qui n'est pas sur le chemin alors que le mot de Christoffel est supposé se rapprocher au maximum de la droite.

2) Propriétés

Figure 6 : propriétés des mots de Christoffel sur l'exemple de la droite de pente 2/3

2/3 = "xxyxy" = "00101"

- "xxyxy" = "x" + "xyx" + "y"
  "yxyxx" = "y" + "xyx" + "x"
  palindrome
- "xxyxy" | "yxyxx"
  symétrie
- "x" ≠ "y"
  différence premier / dernier symbole
- "x" → "0"
  "y" → "1"
  "-x" → "2"
  "-y" → "3"

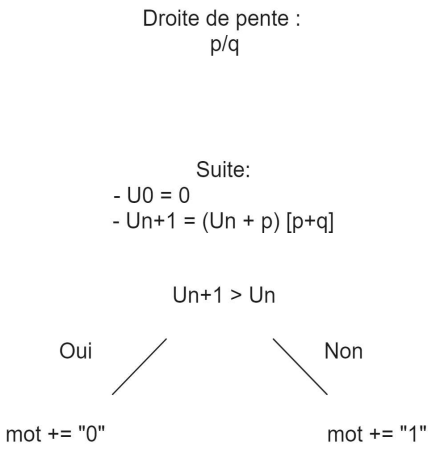


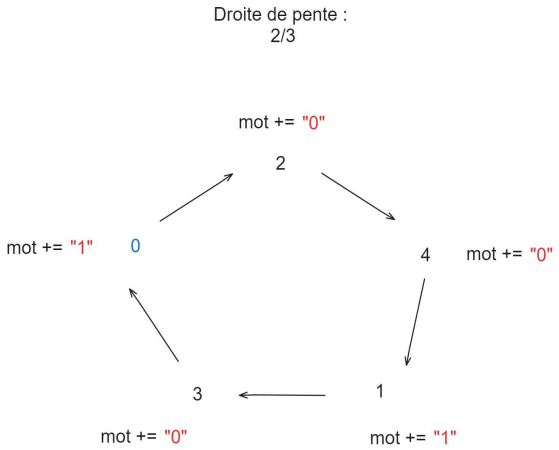
- En retirant le premier et dernier symboles d'un mot de Christoffel on obtient un palindrome.
- Le mot de Christoffel haut est le miroir du mot de Christoffel bas et réciproquement
- Le premier symboles d'un mot de Christoffel est toujours différent du dernier
- On peut généraliser les symboles des mots de Christoffel au reste du plan. On obtient un nouvel alphabet : {0,1,2,3}

3) trouver les mots de Christoffel

i - En obtenir

Figure 7 : définition d'une suite et application sur la droite de pente 2/3





En appliquant la suite définie sur la figure 7 on peut obtenir le mot de Christoffel bas d'une pente p/q.
L'exemple de la figure 7 donne : 2/3 → « 00101 »

ii - En obtenir

Figure 8 :

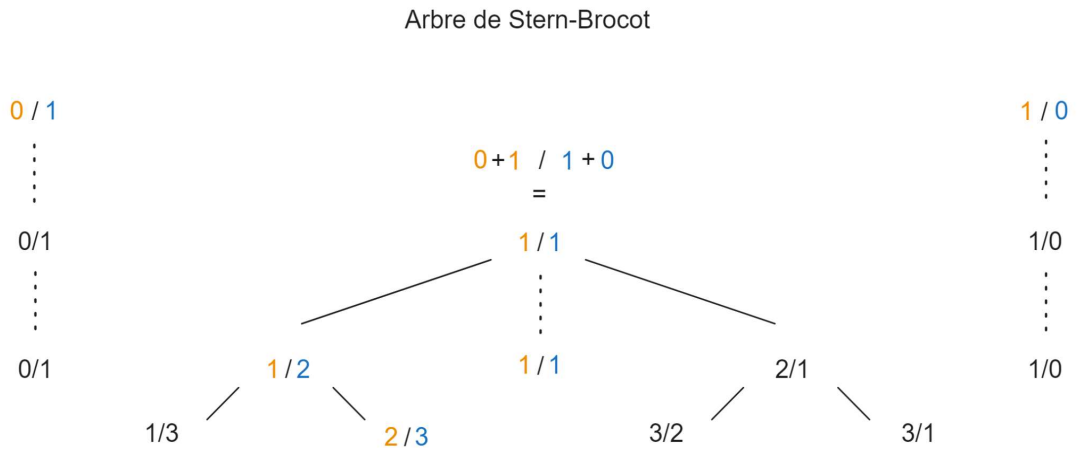
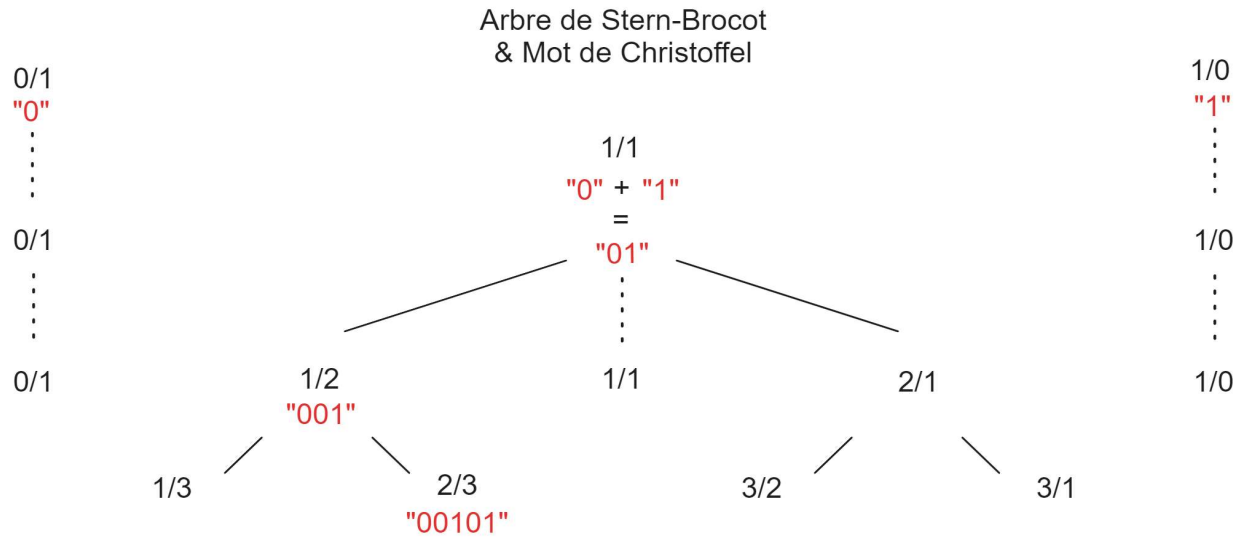


Figure 9 :



L'arbre de Stern-Brocot permet d'obtenir le mot de Christoffel d'une fraction p/q par concaténation des mots de Christoffel des « parents » de la fraction.
On peut alors faire un algorithme qui permet de trouver un mot de Christoffel par un raisonnement dichotomique.

iii – En obtenir

Figure 10 : Fractions continues simples et exemple.
Source : https://math.unice.fr/~walter/L1_Arith/cours2.pdf

$$\frac{p_n}{q_n} = [a_0, a_1, \ldots, a_n] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \; a_{n-1} + \cfrac{1}{a_n}}}.$$

$$\frac{2}{3} = [0,1,2] = 0 + \cfrac{1}{1 + \frac{1}{2}}$$

Une fraction p/q peut s'écrire de la manière décrite sur la figure 10.
On en « extrait » des coefficients entiers qu'on peut utiliser dans la suite définie ci-dessous :

Figure 11 : Suite permettant de trouver un mot de Christoffel en utilisant les coefficients d'une fraction p/q représentée sous la forme montrée en figure 10 & application sur la droite de pente 2/3

p/q = [a0, a1, a2, ... , aN]

Uimpair = "0"
Upair = "1"
i=0

Uimpair = Uimpair + Upair * a(i)
Upair = Uimpair * a(2*i+1) + Upair
i+=2

\+ : opération de concaténation
* : opération de concaténation

2/3 = [0,1,2]

Uimpair = "0"
Upair = "1"

Uimpair = "0" + "1" * 0 = "0"
Upair = "0" * 1 + "1" = "01"
i = 0

Uimpair = "0" + "01" * 2 = "00101"
i = 2

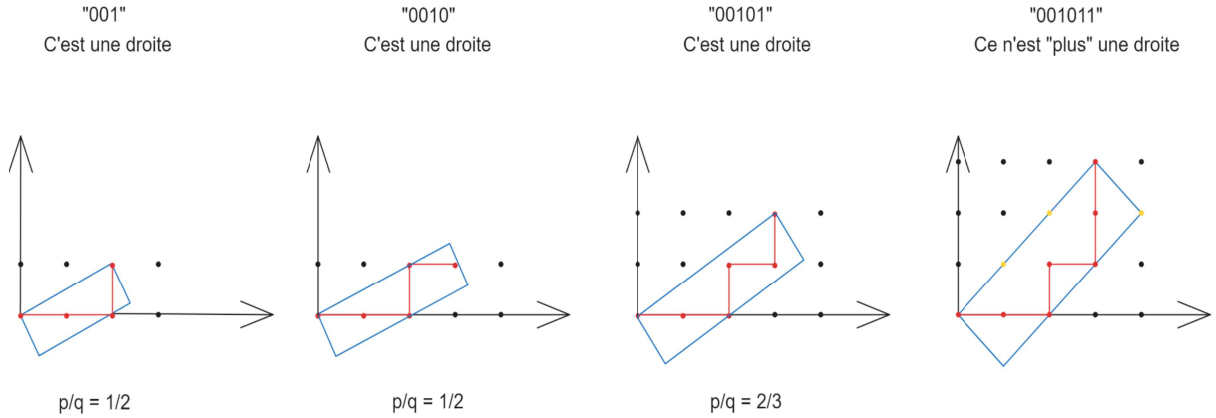
Cette suite permet aussi de trouver le mot de Christoffel bas d'une fraction p/q.

Reconnaissance de droites

1) Reconnaissance dans le quart de plan supérieur droit.

Maintenant que nous savons obtenir un mot de Christoffel à partir d'une droite, nous allons faire l'inverse, c'est-à-dire trouver une pente à partir d'un mot de Christoffel.

Figure 12 : Représentation de l'algorithme de reconnaissance



On donne en paramètre une liste de points entiers obtenus à partir d'un mot.
Le programme « coince » ces points dans une zone (rectangles bleus sur la figure 12) où chaque point entier se trouve forcément sur le chemin du mot de Christoffel. S'il existe des points dans cette zone mais pas sur le chemin cela signifie que le mot ne correspond pas à une droite.
Sinon on prend la pente des côtés les plus longs de la zone et cela nous donne la pente de la droite associée au mot.
Comme nous pouvons maintenant reconnaître des pentes d'un mot composé de « 0 » et de « 1 » il faut maintenant pouvoir le faire avec les symboles « 2 » et « 3 ».

2) Reconnaissance dans le plan

Nous allons effectuer des changements de quadrant : le principe est de convertir un mot fait de {0,1,2,3} en un mot fait de {0,1}, appliquer l'algorithme de reconnaissance sur le nouveau mot afin d'obtenir une pente p'/q' pour finalement reconvertir cette pente en une autre p/q en effectuant quelques changements.

Figure 13 : Changement de quadrant supérieur gauche

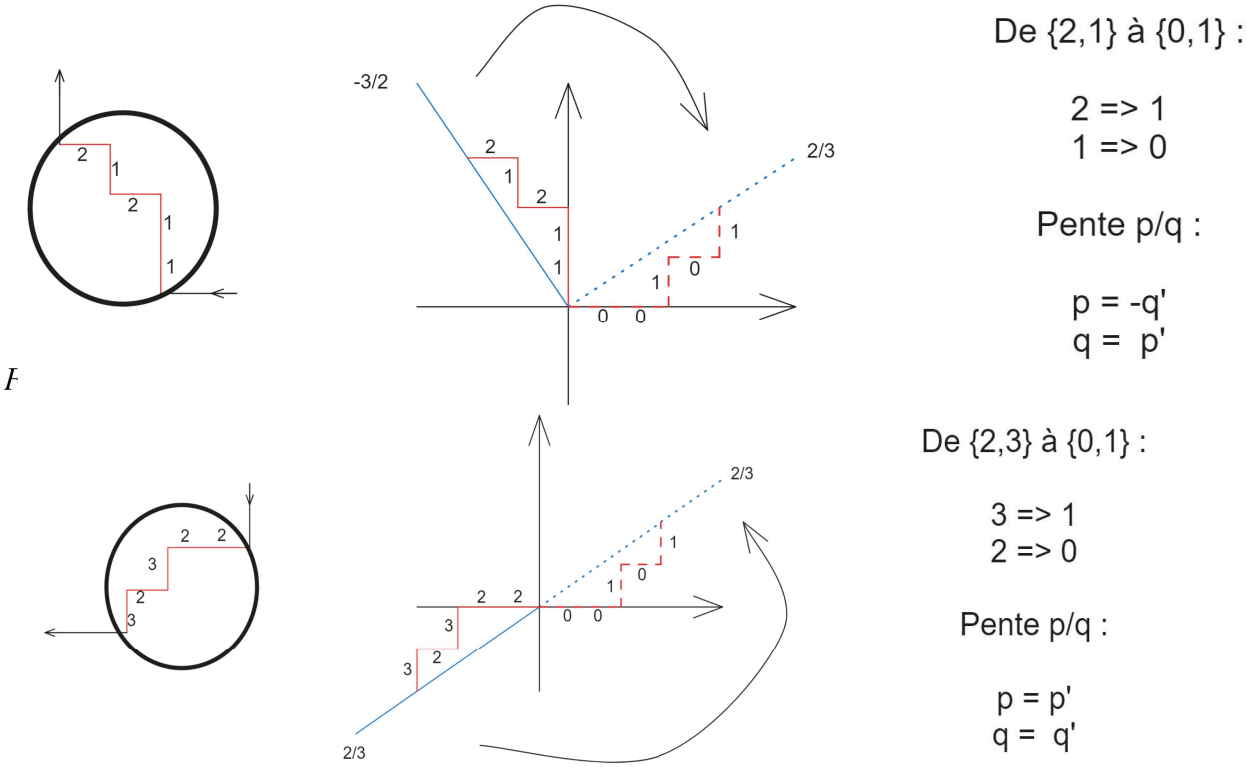
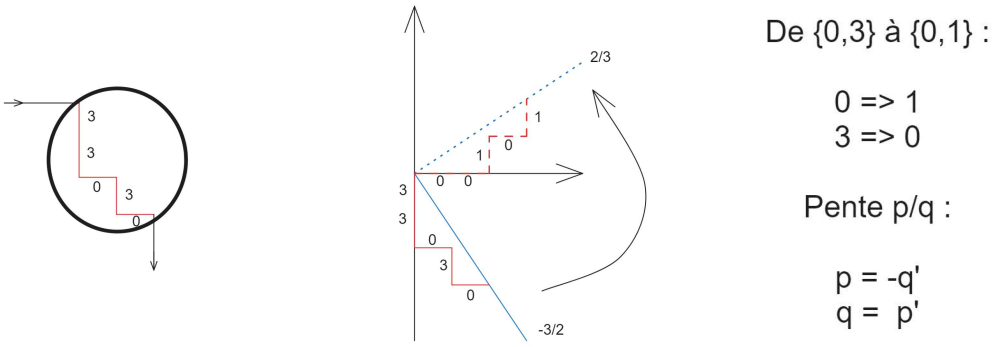


Figure 15 : Changement de quadrant inférieur droit



Puisque l'on sait reconnaître une pente se trouvant dans n'importe quel mot représentant une droite nous allons voir comment trouver toutes les droites d'un mot de contour.

Droites dans un mot de contour.

Figure 16 : Exemple d'un mot de contour

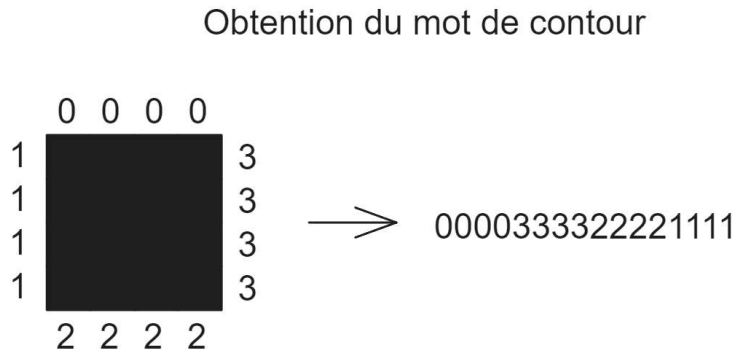
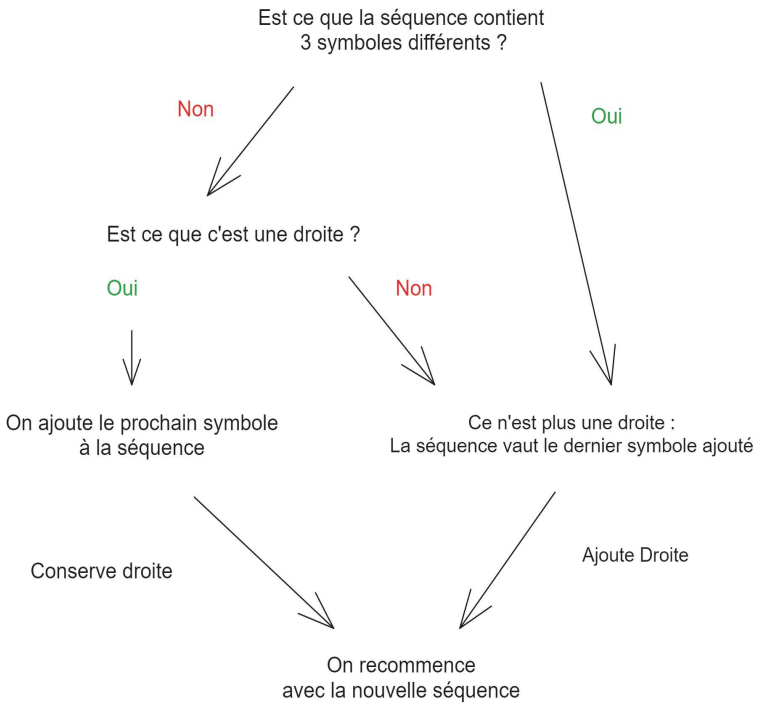


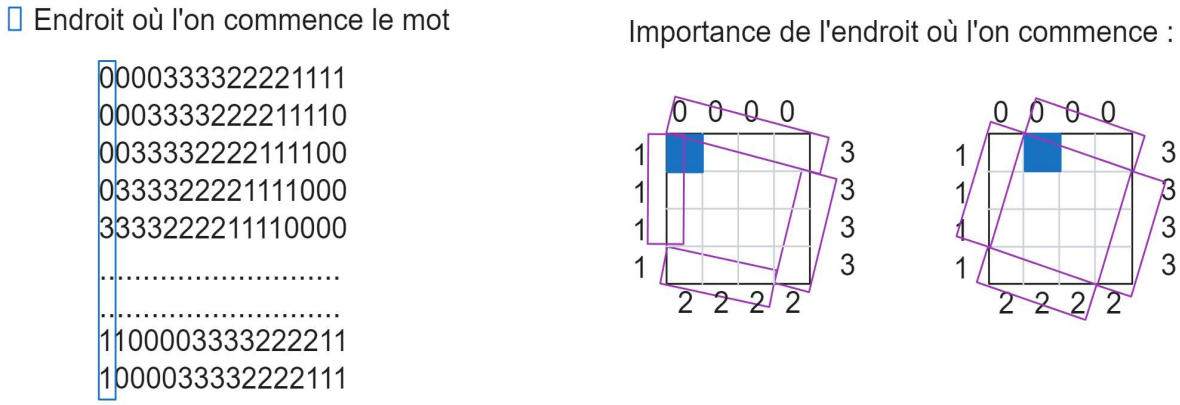
Figure 17 : Logique de l'algorithme de reconnaissance de droites dans un mot de contour.



Une séquence est une chaîne de caractères pouvant contenir des « 0 », « 1 », « 2 » et « 3 ».
Le but est de donner à notre algorithme notre mot de contour avec un mot vide puis en ajoutant un symbole à chaque fois.
Si l'algorithme reconnaît une droite dans la séquence, il sauvegarde la pente associée, sinon il commence une nouvelle séquence au dernier symbole ajouté.

Cependant le fait de commencer à un endroit particulier du mot de contour peut entraîner l'un des problèmes suivants : Le nombres de droites trouvées sera égal au nombre de côtés du polygone ou au nombre de côtés + 1. Les pentes des droites trouvées ne sont pas idéales pour pouvoir reconnaître la forme du polygone.

Figure 18 : représentation de la situation décrite ci-dessus. Le carré bleu indique le pixel de début



Il s'agit donc de tester tous les débuts possibles afin de trouver le « bon » nombre de droites dans le mot de contour et d'avoir toutes les représentations possibles.

Il ne reste plus qu'à obtenir un mot de contour à partir d'une image et nous aurons résolu notre problème car nous pouvons désormais analyser un mot de contour.

Obtenir un mot de contour à partir d'une image

Pour ce faire, nous allons parcours notre image pixel par pixel, de gauche à droite et de haut en bas afin de trouver le premier pixel noir.
Parce qu'on a parcouru notre image de cette manière, on se donnera un vecteur directeur (en rouge sur la figure 19) et un vecteur tangent (en vert sur la figure 19) qui désigneront deux cases et évolueront en fonction de la couleur des cases désignées.

Figure 19 : Mot de contour souhaité et premier pixel noir trouvé avec ses vecteurs et cases désignées

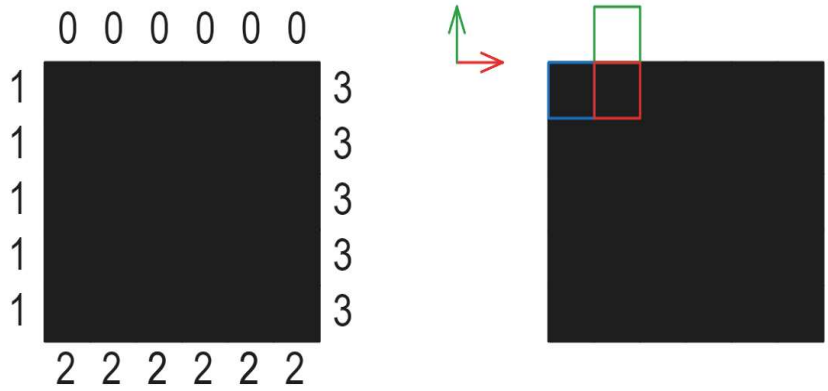
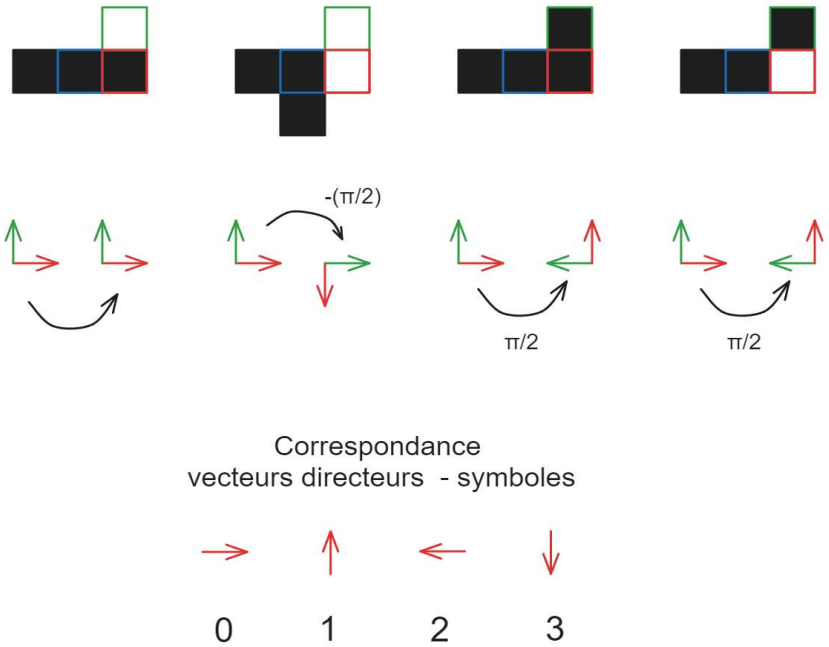


Figure 20 : Cas généraux possibles des couleurs des cases désignés ainsi que les changements de vecteurs a effectué en fonction de ces cas



À chaque situation, on actualise les vecteurs, ajoute au mot le symbole correspondant au nouveau vecteur directeur et change de pixel sur lequel nous nous « trouvons » afin de continuer l'extraction du contour.

L'analyse et l'extraction d'un mot de contour étant expliqué, nous pouvons maintenant associer les différents algorithmes afin de répondre à la problématique : compter le nombre de côté d'un polygone dans une image en comptant le nombre de droites minimales de son mot de contour.

Conclusion

Les prochains travaux à réaliser seraient des travaux :

- d'optimisation : l'algorithme qui regroupe les autres a au moins une complexité O(n^2).
- d'analyse de convexité : il existe un algorithme qui détermine la convexité de la figure à partir de son mot de contour.
- de reconnaissance de formes : puisqu'on a le nombre de côté et les pentes des côtés on pourrait les analyser pour trouver quel polygone est représenté.