

Expériences

Nous avons expérimenté notre code à plusieurs reprises afin de tester ses performances.

Nous avons d'abord testé deux fichiers qui contenaient le code d'un sudoku et d'un morpion (<https://c.developpez.com/telecharger/detail/id/454/Utilisation-de-l-algorithme-Minimax-MinMax>. <https://c.developpez.com/telecharger/detail/id/2951/resolveur-sudoku-simple>).

Ce test a été assez concluant puisque nous avons obtenu un résultat final de 0,97, les fichiers n'étant absolument pas similaire. Les résultats des images pgm sont dans ce cas assez difficile à comprendre. Il est néanmoins possible pour les distances de Dice de voir des pixels plus ou moins foncé en fonction de leur similarité. Nous avons donc testé l'algorithme avec le fichier sudoku que l'on a comparé à lui-même. Le résultat a été beaucoup plus explicite et vraiment intéressant. Nous avons obtenu pour le fichier filtrage.pgm, une image noire avec une diagonale blanche, ce qui correspond bien au résultat recherché. Chaque segment est identique à son homologue dans l'autre fichier et différent des autres. La distance finale n'étant cependant pas satisfaisante pour deux fichiers identiques (parfois autour de 0,55) ce qui nous a forcé de revoir notre code et de transformer les éléments problématiques pour désormais avoir une distance nulle.

Nous avons également testé le fichier avec nos TP du cours de programmation C, supposé similaire bien que pas exactement identique. Le résultat a été satisfaisant puisqu'on obtenait une distance finale faible mais tout de même non nulle.

Enfin un algorithme de calcul de PGCD a été testé de plusieurs manières. Nous l'avons comparé à un autre identique où seul le nom des variables a été changé et la détection de plagiat a été correcte. Puis nous avons modifié la copie du fichier en rajoutant des retours à la ligne et des modifications d'indentations. Ces derniers ont été gérés par l'algorithme, cependant le retour à la ligne crée des segments vides, donc un décalage de la matrice et une augmentation de la distance finale.

La correction de ce bug a été difficile étant donné que la fonction de création de la liste chaînée avait déjà été faite mais nous a permis d'obtenir un algorithme plus performant et plus précis.