

Documentation Projet Complexité

I. Architecture du projet

Les fichiers header sont placés dans le dossier « header ».
Les fichiers C sont à la racine du projet, comme le makefile.

Fonction des fichiers :

- structFichier.h : structure d'un fichier .pgm, composée d'un tableau 2D de valeurs représentant les valeurs des pixels du fichier, cette structure s'appelle PGMValeurs (alias de PGMStructure).
- pgm.h : sert à lire un fichier .pgm et à récupérer sa structure PGMValeurs.
- histogramme.h : construit l'histogramme de la structure PGMValeurs passée en paramètre, il sert aussi à obtenir le seuil entre les pixels blancs et noirs.
- fichierRondelle.h : cherche toutes les rondelles d'un fichier et les modélise, le tableau `int ***rondelles` retourné est donc un tableau 3D dans lequel sont stockées les rondelles, des tableaux 2D.
- traitementRondelle.h : parcourt la rondelle, trouve sa largeur, et si elle a un défaut de surface.

II. Algorithme de construction des rondelles

Pour récupérer et construire les rondelles nous procédons à un scan du fichier. Dès que nous rencontrons un pixel noir (càd supérieur au seuil établi dans histogramme.h), nous testons si nous l'avons déjà rencontré, sinon, c'est que nous avons rencontré la pointe Nord de la rondelle, nous scannons alors cette rondelle en récupérant ses dimensions. Enfin, nous procédons à la construction des rondelles grâce aux dimensions que nous avons récupéré.

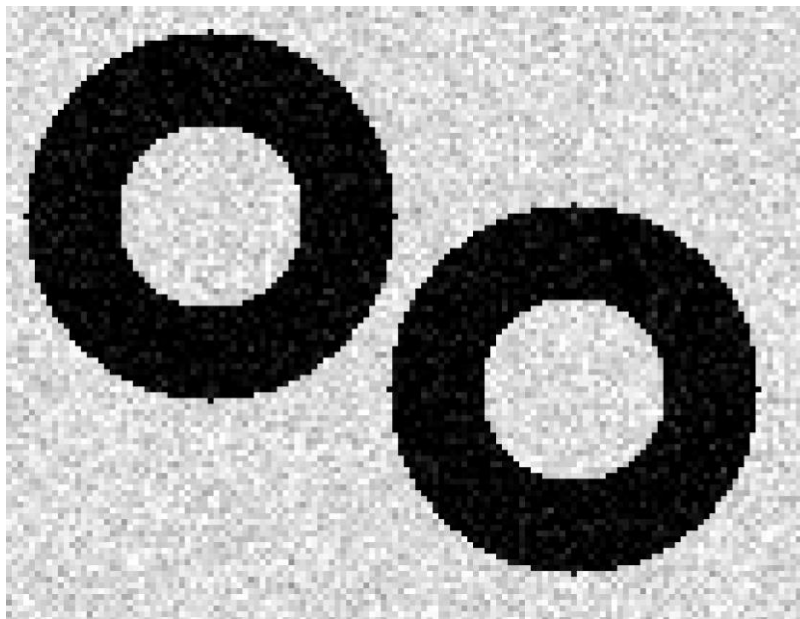
III. Algorithme de recherche de défauts

Pour savoir si une rondelle est défectueuse, nous testons d'abord sa largeur, si elle est inférieure ou supérieure à 17 (c'est la largeur normale des rondelles), c'est que la rondelle a un défaut de dimension. Si la rondelle passe ce test, nous la parcourons et faisons la somme des pixels noirs, si cette somme est supérieure à 2416 (somme des pixels noirs d'une rondelle bien formée) alors la rondelle a un défaut de surface.

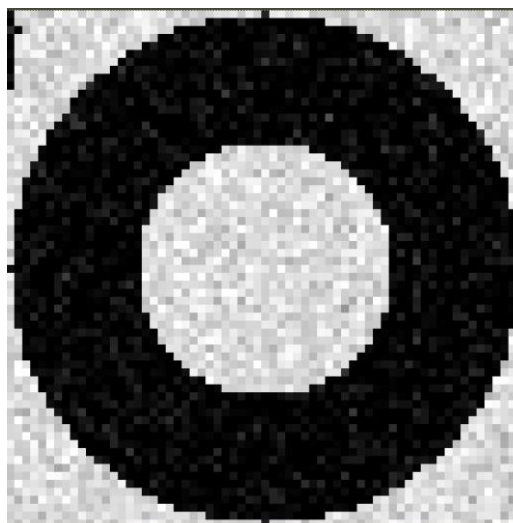
IV. Difficultés et anomalies

Notre principale difficulté aura été la détection de plusieurs rondelles. Nous avons réussi à fixer cela en mémorisant les coins des rondelles, ce qui m'a permis de déterminer si une rondelle a déjà été visitée ou non.

Toutefois, il reste une anomalie due à la somme des pixels noirs, si 2 rondelles sont confondues (cf. captures ci-dessous), alors la somme des pixels est supérieure à 2416, pourtant, la rondelle en possède bien 2416.



Deux rondelles confondues



La rondelle une fois enregistrée

V. Améliorations futures

Pour contrer ce défaut, nous aurions pu mettre en place un autre algorithme qui parcourrait la rondelle (4 boucles for en escalier, cf. capture ci-dessous) et qui testerait si les pixels rencontrés sont supérieurs au seuil blancs-noirs.

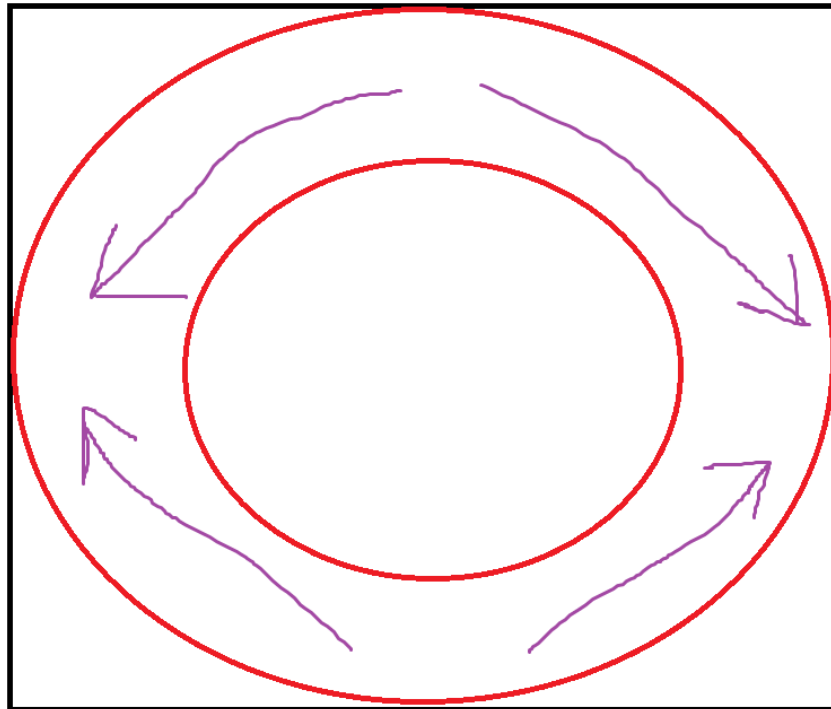


Illustration de l'algorithme décrit plus haut

VI. Précision

Dans resultats.csv, nous avons affiché 3 décimales car notre programme étant trop rapides pour n'en afficher que 2.