

Données Multimédia HAI605I

-Compte Rendu TP2-

Ozgur Dogan - Group A
21811290

Ex1)

Pour tester mon programme j'ai choisi l'image 'lena256.pgm' et j'ai appliqué une valeur de seuillage 70, 100, 120 en taille 256*256. Et le plus intéressant était celui avec la valeur 100.



Lena avec seuillage 100

Pour faire l'érosion je parcours tout l'image et je regarde ces voisins puis s'il a voisin blanc, on transforme ce pixel centrale en blanc sinon on laisse noir. Comme ça on enlève le bruit d'image et il devient plus facile à traiter. Voici mon code et mon output :

```
for (int i=0; i < nH; i++){
    for (int j=0; j < nW; j++){
        for (int k = -1; k < 2; ++k)
        {
            for (int t = -1; t < 2; ++t)
            {
                int a = i+k;
                int b = j+t;
                int indice = a*nW+b;
                if (a < 0 || a >= nH || b < 0 || b >= nW)
                    continue;
                else {
                    if (ImgIn[indice] == 255){
                        ImgOut[i*nW+j] = 255;
                        break;
                    }
                }
            }
        }
    }
}
```



seuillage 100 puis erosion

Ex2)

Pour faire la dilatation j'ai utilisé mon programme précédent mais cette fois si on transforme le pixel central en noire dans le cas s'il a un voisin noir. Et en plus avant on initialise à blanc notre image output.



dilatation

Ex3)

a) Pour faire la fermeture(dilatation+érosion) sur la image, j'ai pris les mêmes fonctions que que j'avais écrit dans exercice précédent et je las ai enchainés. Puisque je fais 2 opération enchainé, j'ai aolué une 2ème tableau avec « allocation_tableau(ImgOut2, OCTET, nTaille); » Comme ça j'ai bouché les trous sur l'image. Puis j'ai testé mon programme avec lena seuillé 100.



fermeture

b) Pour faire la ouverture(érosion+dilatation) sur la image, j'ai pris les fonctions que que j'avais écrit dans exercice précédent et je las ai enchainés. Puisque je fais 2 opération enchainé, j'ai aolué une 2ème tableau avec « allocation_tableau(ImgOut2, OCTET, nTaille); » Comme ça j'ai supprimé les points parasites du fond de l'image. Puis j'ai testé mon programme avec lena seuillé 100.



ouverture

c) Pour faire la fermeture + ouverture sur l'image 'lena'. J'ai exécuté la programme fermeture puis ouverture sur l'image de sortie puis vice versa.



fermeture + ouverture



ouverture + fermeture

d) J'ai exécuté mes programmes enchainement sur la sortie d'image (3 x dilataion puis 6 x erosion puis 3 x dilataion). Comme ça je presque rempli toutes les trous et enlevé les parasites.



3 x d, 6 x e, 3 x d

Ex4) Pour voir la difference sur la 2 images, mon programme prendre 2 image binaire en entrée puis en parcourant tous ses 2 images de même taille ; si leur pixel on la même valeur alors on laisse mon sortie image blanc sinon on rempli en noir pour voir les pixel difference. Pour tester mon programme j'ai utilisé lena avec dilatation et seuillé avec 100 :



Difference.pgm