

## c\hough.c

```
1 // Objectif : utiliser du multithreading pour améliorer la rapidité
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <math.h>
6
7 int COLONNES;
8 int LIGNES;
9 int DISTANCE;
10
11 // taille du fichier. update global variables
12 int* taille(char* filepath){
13     FILE* ptr;
14     char ch;
15     ptr = fopen(filepath, "r");
16     if (NULL == ptr) {
17         printf("file can't be opened \n");
18     }
19     int lignes = 0;
20     int colonnes = 1;
21     int end = 1;
22     while (!feof(ptr)) {
23         ch = fgetc(ptr);
24         if (ch == '\n'){
25             lignes ++;
26             end = 0;
27         }
28         else if(ch == ',' && end) {
29             colonnes ++;
30         }
31     }
32
33     LIGNES = lignes;
34     COLONNES = colonnes;
35     DISTANCE = (int)round(sqrt(COLONNES *COLONNES + LIGNES *LIGNES));
36 }
37
```

```

38 //retourne la matrice
39 int* read_img(char* filepath){
40     int* matrice = malloc((COLONNES * LIGNES) * sizeof(int));
41
42     FILE* ptr;
43     char ch;
44     ptr = fopen(filepath, "r");
45
46     if (NULL == ptr) {
47         printf("file can't be opened \n");
48     }
49     int c = 0;
50     while (!feof(ptr)) {
51         ch = fgetc(ptr);
52         if (feof(ptr)){
53             fclose(ptr);
54             return matrice;
55         }
56         if (ch == '\n'){
57             c++;
58         }
59         else if(ch == ',') {
60             c++;
61         }
62         else if (ch == ' '){}
63         else {
64             //printf("%c\n", ch);
65             matrice[c] *= 10;
66
67             matrice[c] += (int)ch - 48;
68         }
69     }
70 }
71
72 void print_matrice(int* mat){
73     for (int i = 0; i<LIGNES; i++){
74         for (int j = 0; j < COLONNES; j++){
75             printf("%d | ", mat[i*COLONNES + j]);
76         }
77         printf("\n");
78     }

```

```

79 }
80
81 double rho(int x, int y, double theta){
82     return(x * cos(theta) + y * sin(theta));
83 }
84
85 int* espace_hough(int* image, int precision){
86     int* mat_hough = malloc(360 * precision * DISTANCE * sizeof(int));
87     int v_rho;
88     double theta_rad;
89     for (int i = 0; i < 360 * DISTANCE * precision; i++){
90         mat_hough[i] = 0;
91     }
92     for (int y = 0; y < LIGNES; y++){
93         for (int x = 0; x < COLONNES; x++){
94             if (image[y*COLONNES + x] > 250){
95                 for (int theta_deg = 0; theta_deg < 360 * precision ;theta_deg++){
96                     theta_rad = theta_deg / (180.0 * precision) * M_PI;
97                     v_rho = rho(x, y, theta_rad);
98                     if (v_rho > 0){
99                         mat_hough[v_rho * 360 * precision + theta_deg] += 1;
100                     }
101                 }
102             }
103         }
104     }
105     return(mat_hough);
106 }
107
108
109 void print_hough(int* mat){
110     for (int i = 0; i < 360; i++){
111         for (int j = 0; j < DISTANCE; j++){
112             printf("%d | ", mat[j*360 + i]);
113         }
114         printf("\n");
115     }
116 }
117
118 void save_file(int* mat, int precision){
119     FILE *ptr;

```

```

120 ptr = fopen("./out.txt","w");
121
122 for (int j = 0; j < DISTANCE; j++){
123     for (int i = 0; i<360 * precision; i++){
124         fprintf(ptr, "%d", mat[j*360 * precision + i]);
125         if (i != 360 * precision - 1){
126             fprintf(ptr, "%s", ",");
127         }
128     }
129     fprintf(ptr, "%s", "\n");
130 }
131
132 fclose(ptr);
133
134 }
135
136
137 int main()
138 {
139     int precision = 100;
140     printf("precision : %d\n", precision);
141     char* filepath = "./in.txt";
142     taille(filepath);
143     printf("Lignes : %d\nColonnes : %d\nnombre de pixels : %d\nDistance : %d\n", LIGNES, COLONNES, (LIGNES)*(COLONNES), DISTANCE);
144     int* image = read_img(filepath);
145     //print_matrice(image);
146     int* espace = espace_hough(image, precision);
147     //print_hough(espace);
148
149     save_file(espace, precision);
150     return 0;
151 }
152
153

```