

```

1 /* blurGood.c 使用メモリ小&高速処理 */
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <malloc.h>
5 #include <time.h>
6 #include <sys/time.h>
7 #define M 1024
8 #define N 1024
9 int main( void )
10 {
11     FILE      *fpr,*fpw;
12     unsigned char  *ipt, *ipc, *ipb, *TMP;
13     double      sum;
14     int         bip, i, j;
15     double      ts, te, etimeClock, etimeGettimeofday;
16     clock_t     start, finish;
17
18     double gettimeofday_sec()
19     {
20     struct timeval tv;
21     gettimeofday(&tv, NULL);
22     return tv.tv_sec + (double)tv.tv_usec*1e-6;
23     }
24
25     ipt = (unsigned char *)calloc( M, sizeof(unsigned char) );
26     ipc = (unsigned char *)calloc( M, sizeof(unsigned char) );
27     ipb = (unsigned char *)calloc( M, sizeof(unsigned char) );
28     if (ipt==NULL || ipc==NULL || ipb==NULL) {
29         fprintf(stderr, "ERROR: cannot allocate memory¥n");
30         exit(EXIT_FAILURE);
31     }
32
33     fpr = fopen("myareaNew.gray","rb");
34     fpw = fopen("blurGood.gray","wb");
35     if (fpr==NULL || fpw==NULL) {
36         fprintf(stderr, "ERROR: cannot open file¥n");
37         exit(EXIT_FAILURE);
38     }
39
40     start=clock();
41     ts = gettimeofday_sec();
42     fread(ipt,M*sizeof(unsigned char),1,fpr); /* top */
43     fread(ipc,M*sizeof(unsigned char),1,fpr); /* center */
44     fread(ipb,M*sizeof(unsigned char),1,fpr); /* bottom */
45
46     /* 3 X 3 Window Operation */
47     for (j=1; j < N-1; j++){
48         sum =0.000;
49         for (i=1; i < M-1; i++){
50             sum =ipt[i-1]+ipc[i-1]+ipb[i-1];
51             sum+=ipt[i ]+ipc[i ]+ipb[i ];
52             sum+=ipt[i+1]+ipc[i+1]+ipb[i+1];
53             sum = sum/9.0;
54             bip = (int) (sum + 0.5);
55             if (bip > 255) bip = 255;
56             if (bip < 0 ) bip = 0;
57             fputc(bip,fpw);
58         }
59         TMP = ipt;
60         ipt = ipc;
61         ipc = ipb;
62         ipb = TMP;
63         fread(ipb,M*sizeof(unsigned char),1,fpr);
64     }
65     te = gettimeofday_sec();
66     finish=clock();

```

```
67
68     etimeGettimeofday = te - ts;
69     etimeClock=(double) (finish-start)/CLOCKS_PER_SEC;
70     printf("elapse time: %lf seconds (%lf precise sec.)\n", etimeClock, etimeGettimeofday);
71
72     fclose(fpr);
73     fclose(fpw);
74     return EXIT_SUCCESS;
75 }
```