# 分形算法Mandelbrot并行化

将下列分形算法用OpenMP并行化，并比较和分析1，2，4，8 CPU（核）的扩展性和加速性。

#include <stdlib.h>

#include <GL/glut.h>

/\* Defaut data via command line \*/

/\* Can enter other values via command line arguments \*/

#define CENTERX -0.5

#define CENTERY 0.5

#define HEIGHT 0.5

#define WIDTH 0.5

#define MAX\_ITER 100

/\* N x M array to be generated \*/

#define N 500

#define M 500

float height = HEIGHT; /\* size of window in complex plane \*/

float width = WIDTH;

float cx = CENTERX; /\* center of window in complex plane \*/

float cy = CENTERY;

int max = MAX\_ITER; /\* number of interations per point \*/

int n=N;

int m=M;

/\* Use unsigned bytes for image \*/

GLubyte image[N][M];

/\* Complex data type and complex add, mult, and magnitude functions \*/

/\* Probably not worth overhead \*/

typedef float complex[2];

void add(complex a, complex b, complex p)

{

p[0]=a[0]+b[0];

p[1]=a[1]+b[1];

}

void mult(complex a, complex b, complex p)

{

p[0]=a[0]\*b[0]-a[1]\*b[1];

p[1]=a[0]\*b[1]+a[1]\*b[0];

}

float mag2(complex a)

{

return(a[0]\*a[0]+a[1]\*a[1]);

}

void form(float a, float b, complex p)

{

p[0]=a;

p[1]=b;

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glDrawPixels(n,m,GL\_COLOR\_INDEX, GL\_UNSIGNED\_BYTE, image);

}

void myReshape(int w, int h)

{

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

if (w <= h)

gluOrtho2D(0.0, 0.0, (GLfloat) n, (GLfloat) m\* (GLfloat) h / (GLfloat) w);

else

gluOrtho2D(0.0, 0.0, (GLfloat) n \* (GLfloat) w / (GLfloat) h,(GLfloat) m);

glMatrixMode(GL\_MODELVIEW);

display();

}

void myinit()

{

float redmap[256], greenmap[256],bluemap[256];

int i;

glClearColor (1.0, 1.0, 1.0, 1.0);

gluOrtho2D(0.0, 0.0, (GLfloat) n, (GLfloat) m);

/\* Define pseudocolor maps, ramps for red and blue,

random for green \*/

for(i=0;i<256;i++)

{

redmap[i]=i/255.;

greenmap[i]=drand48();

bluemap[i]=1.0-i/255.;

}

glPixelMapfv(GL\_PIXEL\_MAP\_I\_TO\_R, 256, redmap);

glPixelMapfv(GL\_PIXEL\_MAP\_I\_TO\_G, 256, greenmap);

glPixelMapfv(GL\_PIXEL\_MAP\_I\_TO\_B, 256, bluemap);

}

main(int argc, char \*argv[])

{

int i, j, k;

float x, y, v;

complex c0, c, d;

if(argc>1) cx = atof(argv[1]); /\* center x \*/

if(argc>2) cy = atof(argv[2]); /\* center y \*/

if(argc>3) height=width=atof(argv[3]); /\* rectangle height and width \*/

if(argc>4) max=atoi(argv[4]); /\* maximum iterations \*/

for (i=0; i<n; i++) for(j=0; j<m; j++)

{

/\* starting point \*/

x= i \*(width/(n-1)) + cx -width/2;

y= j \*(height/(m-1)) + cy -height/2;

form(0,0,c);

form(x,y,c0);

/\* complex iteration \*/

for(k=0; k<max; k++)

{

mult(c,c,d);

add(d,c0,c);

v=mag2(c);

if(v>4.0) break; /\* assume not in set if mag > 4 \*/

}

/\* assign gray level to point based on its magnitude \*/

if(v>1.0) v=1.0; /\* clamp if > 1 \*/

image[i][j]=255\*v;

}

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB );

glutInitWindowSize(N, M);

glutCreateWindow("mandlebrot");

myinit();

glutReshapeFunc(myReshape);

glutDisplayFunc(display);

glutMainLoop();

}

算法中用到的OpenGL工具（glut库）在如下地址参考和下载。

http://www.its.monash.edu.au/staff/systems/linux/technical/gluthome.html

http://www.opengl.org/resources/libraries/glut/glut\_downloads.php