Lab Report

Week 9

Jitendra Kumar , 1401CS19

03/04/2017

Title

- Implement Hidden Surface removal for a CUBE.
  1). OpenGL

Procedure

OpenGL

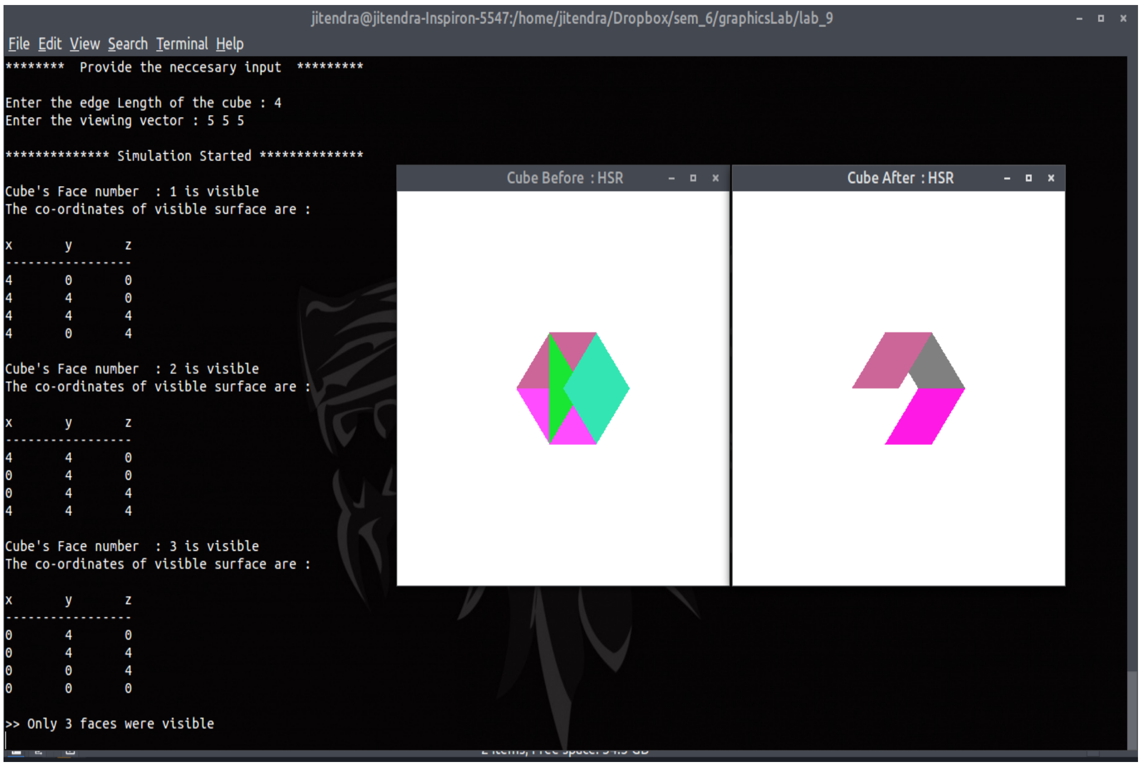
1). Choose N vertices of a CUBE and a viewing vector, then apply the algorithm described below :

- Create a C file and name it as *hsr.c*.
- Following is the final code :

```
#include "GL/glut.h"
#include "GL/gl.h"
#include <math.h>
#include <stdio.h>

int side = 0; int flag = 0;
float vx,vy,vz; int count =0; float a[2][3];
int calculate()
{
        float cross_product[3];
        cross_product[0] = a[0][1]*a[1][2] - a[0][2]*a[1][1];
        cross_product[1] = a[0][2]*a[1][0] - a[0][0]*a[1][2];
        cross_product[2] = a[0][0]*a[1][1] - a[0][1]*a[1][0];

        int dot_product = vx * cross_product[0] + vy * cross_product[1] + vz * cross_product[2];
        if(dot_product > 0)
        {
                printf("Cube's Face number  : %d is visible\n", count);
                return 1;
        }
        return 0;
}
void display_function_after()
{
        printf("\n************** Simulation Started **************\n\n");
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glClear(GL_COLOR_BUFFER_BIT);
        glMatrixMode(GL_MODELVIEW);
        glTranslatef(0.0, 0.0, -1.0);
        glRotatef(-45, 0.0, 1.0, 0.0);
        glRotatef(-45, 0.0, 0.0, 1.0);
        int vertices[6][4][3] =
        {
                { {side , 0 , 0} , {side , side , 0} , {side , side , side} , {side , 0 , side} },
                { {side , side , 0}, {0, side , 0} , {0 , side , side} , {side , side , side} },
                { {0 , side , 0}  , {0 , side , side}, {0 , 0 , side},{0 , 0 , 0} },
                { {0, 0, side}, {0 , 0 , 0}, {side, 0, 0}, {side , 0, side} },
                { {side , 0, side } , {side , side , side} , {0, side, side} , {side, 0, side}},
                { {side , 0 , 0} , {0, 0, 0} , {0, side, 0} , {side, side, 0}}
        };
        int i= 0 ; int j = 0 ;
        for(i = 0 ; i < 6 ; i++)
        {
                count = i+1;
                for(j = 0 ; j < 3 ; j++)
                {
                        a[0][j] = vertices[i][3][j] - vertices[i][0][j];
                        a[1][j] = vertices[i][0][j] - vertices[i][1][j];
                }
                if(calculate())
                {
                        glBegin(GL_QUADS);
                        if(flag == 0) glColor3f(1.0, 0.1, 0.9);
                        else if(flag == 1) glColor3f(0.5, 0.5, 0.5);
                        else if(flag == 2) glColor3f(0.8, 0.4, 0.6);
                        else if(flag == 3) glColor3f(1.0, 0.3, 1.0);
                        else if(flag == 4) glColor3f(0.1, 0.9, 0.2);
                        else               glColor3f(0.2, 0.9, 0.7);

                        printf("The co-ordinates of visible surface are : \n\n");
                        printf("x\ty\tz\n");
                        printf("-----------------\n");
                        printf("%d\t%d\t%d\n",vertices[i][0][0],vertices[i][0][1],vertices[i][0][2]);
                        printf("%d\t%d\t%d\n",vertices[i][1][0],vertices[i][1][1],vertices[i][1][2]);
                        printf("%d\t%d\t%d\n",vertices[i][2][0],vertices[i][2][1],vertices[i][2][2]);
                        printf("%d\t%d\t%d\n",vertices[i][3][0],vertices[i][3][1],vertices[i][3][2]);
                        printf("\n");
                        glVertex3f(vertices[i][0][0]/10.0f, vertices[i][0][1]/10.0f, vertices[i][0][2]/10.0f);
                        glVertex3f(vertices[i][1][0]/10.0f, vertices[i][1][1]/10.0f, vertices[i][1][2]/10.0f);
                        glVertex3f(vertices[i][2][0]/10.0f, vertices[i][2][1]/10.0f, vertices[i][2][2]/10.0f);
                        glVertex3f(vertices[i][3][0]/10.0f, vertices[i][3][1]/10.0f, vertices[i][3][2]/10.0f);
                        flag++;
                        glEnd();
                }
        }
        printf(">> Only %d faces were visible\n",flag);
        printf("**************  Simulation Ended  **************\n");
        glFlush();
        glutSwapBuffers();
}
void display_function_Before()
{
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glClear(GL_COLOR_BUFFER_BIT);
        glMatrixMode(GL_MODELVIEW);
        glTranslatef(0.0, 0.0, -1.0);
        glRotatef(-45, 0.0, 1.0, 0.0);
        glRotatef(-45, 0.0, 0.0, 1.0);
        int vertices[6][4][3] =
        {
                { {side , 0 , 0} , {side , side , 0} , {side , side , side} , {side , 0 , side} },
                { {side , side , 0}, {0, side , 0} , {0 , side , side} , {side , side , side} },
                { {0 , side , 0}  , {0 , side , side}, {0 , 0 , side},{0 , 0 , 0} },
                { {0, 0, side}, {0 , 0 , 0}, {side, 0, 0}, {side , 0, side} },
                { {side , 0, side } , {side , side , side} , {0, side, side} , {side, 0, side}},
                { {side , 0 , 0} , {0, 0, 0} , {0, side, 0} , {side, side, 0}}
        };
        int i = 0 ; int flag=0;
        for(i = 0 ; i < 6 ; i++)
        {
                glBegin(GL_QUADS);
                if(flag == 0) glColor3f(1.0, 0.1, 0.9);
                else if(flag == 1) glColor3f(0.5, 0.5, 0.5);
                else if(flag == 2) glColor3f(0.8, 0.4, 0.6);
                else if(flag == 3) glColor3f(1.0, 0.3, 1.0);
                else if(flag == 4) glColor3f(0.1, 0.9, 0.2);
                else               glColor3f(0.2, 0.9, 0.7);
                glVertex3f(vertices[i][0][0]/10.0f, vertices[i][0][1]/10.0f, vertices[i][0][2]/10.0f);
                glVertex3f(vertices[i][1][0]/10.0f, vertices[i][1][1]/10.0f, vertices[i][1][2]/10.0f);
                glVertex3f(vertices[i][2][0]/10.0f, vertices[i][2][1]/10.0f, vertices[i][2][2]/10.0f);
                glVertex3f(vertices[i][3][0]/10.0f, vertices[i][3][1]/10.0f, vertices[i][3][2]/10.0f);
                flag++;
                glEnd();
        }
        glFlush();
}
int main(int argc, char *argv[])
{
        printf("********  Provide the neccesary input  *********\n\n");
        printf("Enter the edge Length of the cube : "); scanf("%d",&side);
        printf("Enter the viewing vector : ");          scanf("%f%f%f",&vx,&vy,&vz);
        glutInit(&argc,argv);
        glutInitDisplayMode(GLUT_RGB);
        glutInitWindowSize(400, 400);
        gluOrtho2D(-200, 200, -200, 200);
        glutCreateWindow("Cube Before  : HSR");
        glutDisplayFunc(display_function_Before);
        glutCreateWindow("Cube After  : HSR");
        glutDisplayFunc(display_function_after);
        glutMainLoop();
        return 0;
}
```

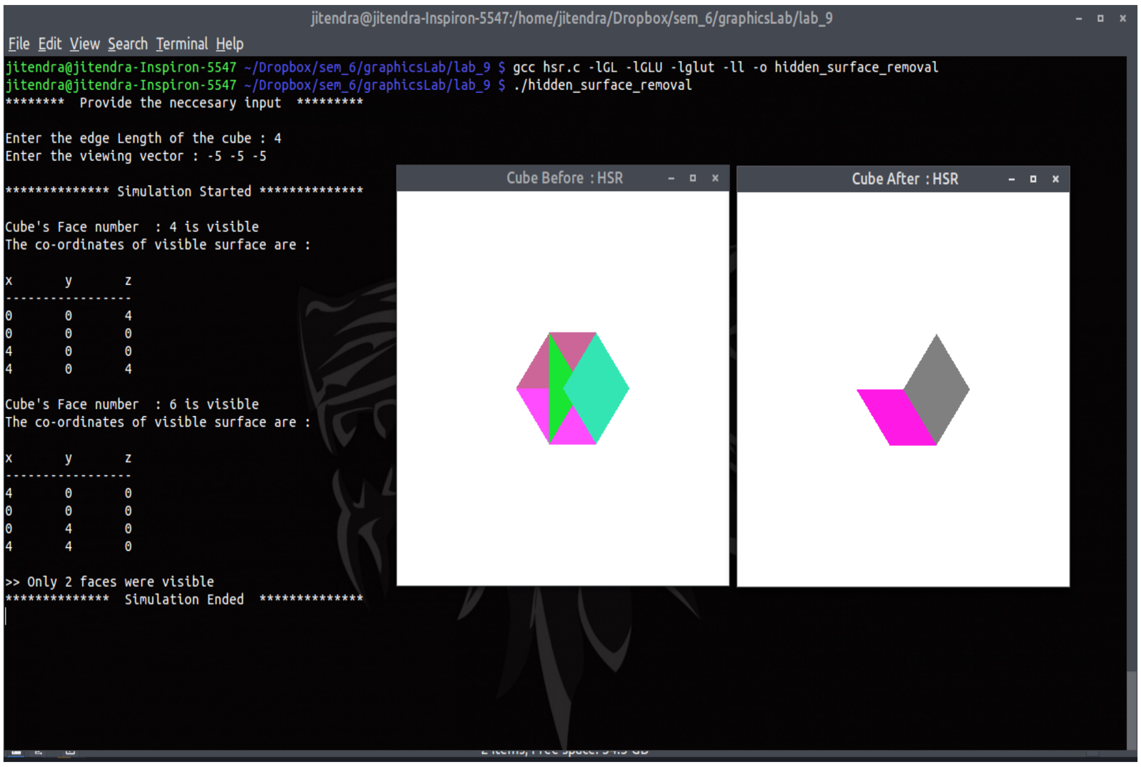
- Compile and run the executable file in terminal by typing in the following commands :
  (a) *gcc hsr.c -lGL -lGLU -lglut -ll -o hsr*
  (b) *./hsr*

Output


jitendra@jitendra-Inspiron-5547:/home/jitendra/Dropbox/sem_6/graphicsLab/lab_9
File Edit View Search Terminal Help
Cube Before : HSR
Cube After : HSR

```
********  Provide the neccesary input  *********

Enter the edge Length of the cube : 4
Enter the viewing vector : 5 5 5

************** Simulation Started **************

Cube's Face number  : 1 is visible
The co-ordinates of visible surface are :

x       y       z
----------------
4       0       0
4       4       0
4       4       4
4       0       4

Cube's Face number  : 2 is visible
The co-ordinates of visible surface are :

x       y       z
----------------
4       4       0
0       4       0
0       4       4
4       4       4

Cube's Face number  : 3 is visible
The co-ordinates of visible surface are :

x       y       z
----------------
0       4       0
0       4       4
0       0       4
0       0       0

>> Only 3 faces were visible
```

FIGURE 1 – openGL output


jitendra@jitendra-Inspiron-5547:/home/jitendra/Dropbox/sem_6/graphicsLab/lab_9
File Edit View Search Terminal Help
Cube Before : HSR
Cube After : HSR

```
jitendra@jitendra-Inspiron-5547 ~/Dropbox/sem_6/graphicsLab/lab_9 $ gcc hsr.c -lGL -lGLU -lglut -ll -o hidden_surface_removal
jitendra@jitendra-Inspiron-5547 ~/Dropbox/sem_6/graphicsLab/lab_9 $ ./hidden_surface_removal
********  Provide the neccesary input  *********

Enter the edge Length of the cube : 4
Enter the viewing vector : -5 -5 -5

************** Simulation Started **************

Cube's Face number  : 4 is visible
The co-ordinates of visible surface are :

x       y       z
----------------
0       0       4
0       0       0
4       0       0
4       0       4

Cube's Face number  : 6 is visible
The co-ordinates of visible surface are :

x       y       z
----------------
4       0       0
0       0       0
0       4       0
4       4       0

>> Only 2 faces were visible
**************  Simulation Ended  **************
```

Figure 2 – openGL output