Lab Report : 06

Title: Computer Graphics Lab
Course code: CSE-304
3rd Year 1st Semester

Date of Submission: **20..08.2023**

Submitted to-

Prof. Dr. Mohammad Shorif Uddin *Professor* *Department of Computer Science and Engineering* *Jahangirnagar University* *Savar, Dhaka-1342*	**Dr. Morium Akther** *Associate Professor* *Department of Computer Science and Engineering* *Jahangirnagar University* *Savar, Dhaka-1342*

Sl	Class Roll	Registration Number	Name
01	388	20200650758	Md.Tanvir Hossain Saon

Experiment Name: Implementation of Liang Barsky Line Clipping algorithm.

Introduction:

The Liang-Barsky Line Clipping algorithm is a seminal technique in the realm of computer graphics. Developed by You-Dong Liang and Brian A. Barsky in the early 1980s, this algorithm harnesses the power of parametric equations to adeptly identify and showcase only those segments of a line that intersect a designated rectangular viewing region. As opposed to traditional line clipping methods which may rely heavily on iterative intersection checks, the Liang-Barsky approach optimizes this process, ensuring both computational efficiency and precise results. Its innovative method emphasizes not only on the immediacy of discarding line portions that lie outside the viewing frame but also on maintaining the integrity and accuracy of the visible line segments within the frame. This nuanced balance has solidified its reputation as a cornerstone algorithm in graphical rendering processes.

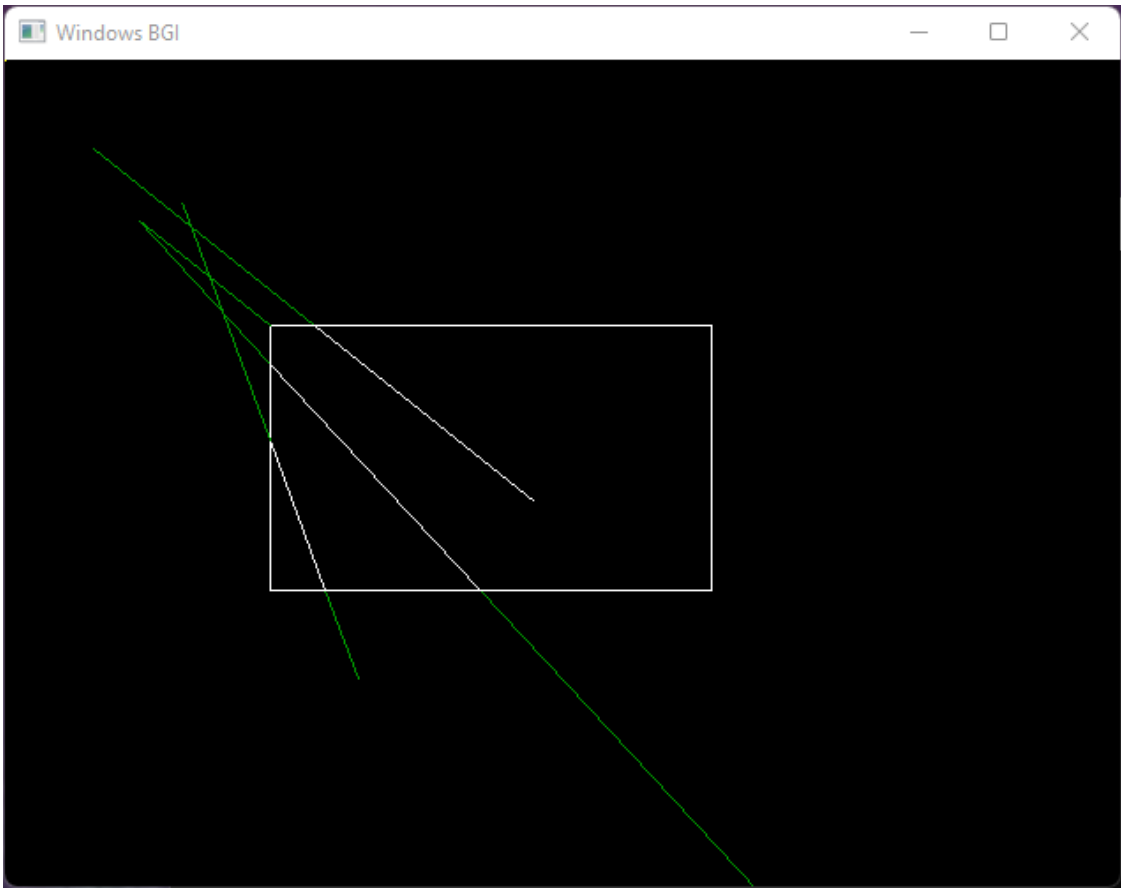
Source Code:

```
#include <iostream>
#include <graphics.h>
using namespace std;
void drawColoredLine(int x1, int y1, int x2, int y2, int color)
{
   setcolor(color);
   line(x1, y1, x2, y2);
   setcolor(WHITE);
}
void liangBarsky(int x1, int y1, int x2, int y2, int xmin, int ymin, int xmax, int ymax)
{
   int p[4], q[4];
   int dx = x2 - x1, dy = y2 - y1;
   p[0] = -dx;
   q[0] = x1 - xmin;
   p[1] = dx;
   q[1] = xmax - x1;
   p[2] = -dy;
   q[2] = y1 - ymin;
   p[3] = dy;
   q[3] = ymax - y1;
   float u1 = 0, u2 = 1;
   for (int i = 0; i < 4; i++)
   {
      if (p[i] == 0)
      {
```

```
            if (q[i] < 0)
            {
                return;
            }
        }
        else
        {
            float t = (float)q[i] / p[i];
            if (p[i] < 0)
            {
                u1 = max(u1, t);
            }
            else
            {
                u2 = min(u2, t);
            }
        }
    }
    if (u1 > u2)
    {
        return;
    }
    int clippedX1 = x1 + u1 * dx;
    int clippedY1 = y1 + u1 * dy;
    int clippedX2 = x1 + u2 * dx;
    int clippedY2 = y1 + u2 * dy;
    drawColoredLine(x1, y1, clippedX1, clippedY1, GREEN);
    drawColoredLine(clippedX1, clippedY1, clippedX2, clippedY2, WHITE);
    drawColoredLine(clippedX2, clippedY2, x2, y2, GREEN);
}
int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int xmin = 150, ymin = 150, xmax = 400, ymax = 300;
    rectangle(xmin, ymin, xmax, ymax);
    liangBarsky(50, 50, 300, 250, xmin, ymin, xmax, ymax);
    liangBarsky(150,150,75,90,xmin,ymin,xmax,ymax);
    liangBarsky(200,350,100,80,xmin,ymin,xmax,ymax);
    liangBarsky(500,550,75,90,xmin,ymin,xmax,ymax);
```

```
    getch();
    delay(50000);
    closegraph();
    return 0;
}
```

Output:



Discussion:

The Liang-Barsky Line Clipping algorithm efficiently clips lines to a rectangular window using basic arithmetic operations. It evaluates a line segment against the window's edges, employing the line's parametric equation to pinpoint potential intersections; this determines if the line is within, outside, or intersecting the window, and then draws the relevant portion of the line accordingly. This method, prevalent in computer graphics, is favored for its speed and simplicity over other clipping algorithms.