Lab Report: 06

Title: Liang–Barsky Line clipping Algorithm

Course title: Computer Graphics Laboratory

Course code: CSE-304

3rd Year 1st Semester Examination 2022

Date of Submission: 13/08/2023

Submitted to-

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

Class Roll	Exam Roll	Name
386	202198	Sunirmol Mollik

Lab Report Title: Understanding Liang–Barsky Line clipping Algorithm

and solving a problem based on this Algorithm.

Introduction:

The Liang-Barsky Line Clipping Algorithm is a technique used to clip a line segment against a rectangular window or any other convex polygon. It efficiently determines the portion of the line that lies within the given clipping region. This algorithm is particularly useful for computer graphics and image processing applications where you want to draw or display only the visible portion of a line.

The Liang-Barsky algorithm avoids the computation of intersection points and instead directly calculates the parameter values of the endpoints of the visible portion of the line segment. This makes it more efficient compared to other line clipping algorithms that require intersection point calculations.

Problem statement:

Let R be the rectangular window whose lower left-hand is at L(-3,1) and upoper right hand corner is at R(2,6)Find the region end points for several points. The points are:

(-4,7),(-2,10)

(-4,2),(-1,7)

(-1,5),(3,8)

(-2,3), (1,2)

(1,-2) ,(3,3)

Source Code:

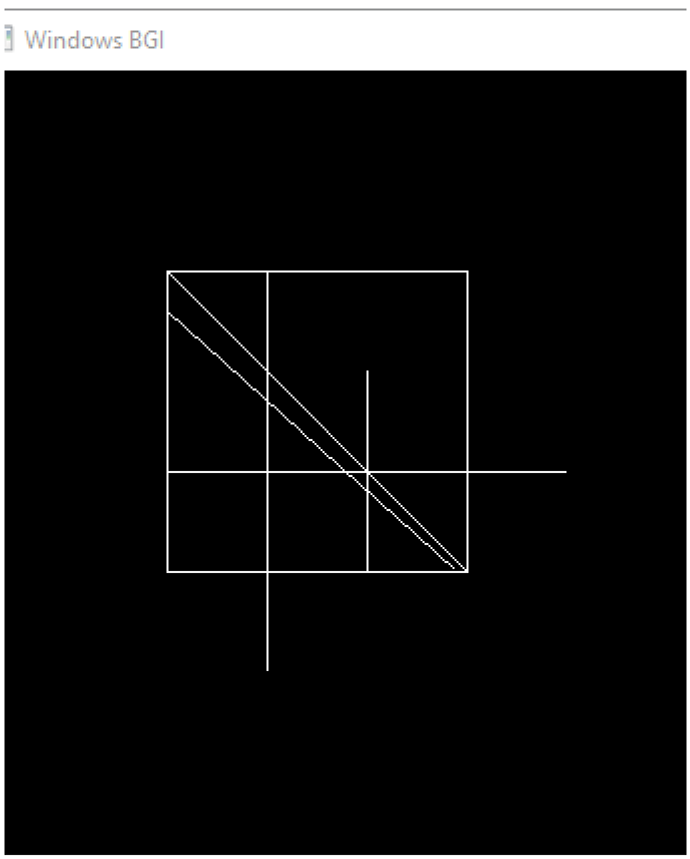
```
#include <bits/stdc++.h>
#include <graphics.h>
#include <math.h>
#include <dos.h>
using namespace std;

int main()
{
    int i, gd = DETECT, gm;
    int x1[5], y1[5], x2[5], y2[5], xmin, xmax, ymin, ymax, xx1, xx2, yy1, yy2, dx, dy;
    float t1, t2, p[4], q[4], temp;
```

```
// Initialize five pairs of points
x1[0] = 100;
y1[0] = 120;
x2[0] = 300;
y2[0] = 300;

x1[1] = 50;
y1[1] = 200;
x2[1] = 400;
y2[1] = 200;

x1[2] = 150;
y1[2] = 50;
x2[2] = 150;
y2[2] = 400;

x1[3] = 50;
y1[3] = 50;
x2[3] = 350;
y2[3] = 350;

x1[4] = 200;
y1[4] = 150;
x2[4] = 200;
y2[4] = 400;

xmin = 100;
ymin = 100;
xmax = 250;
ymax = 250;

initgraph(&gd, &gm, "c:\\turboc3\\bgi");
rectangle(xmin, ymin, xmax, ymax);

for (int j = 0; j < 5; j++)
{
  dx = x2[j] - x1[j];
  dy = y2[j] - y1[j];
  p[0] = -dx;
  p[1] = dx;
  p[2] = -dy;
  p[3] = dy;
  q[0] = x1[j] - xmin;
  q[1] = xmax - x1[j];
  q[2] = y1[j] - ymin;
  q[3] = ymax - y1[j];
  for (i = 0; i < 4; i++)
  {
    if (p[i] == 0)
    {
```

```
        cout << "line is parallel to one of the clipping boundary";
        if (q[i] >= 0)
        {
          if (i < 2)
          {
            if (y1[j] < ymin)
            {
              y1[j] = ymin;
            }
            if (y2[j] > ymax)
            {
              y2[j] = ymax;
            }
            line(x1[j], y1[j], x2[j], y2[j]);
          }
          if (i > 1)
          {
            if (x1[j] < xmin)
            {
              x1[j] = xmin;
            }
            if (x2[j] > xmax)
            {
              x2[j] = xmax;
            }
            line(x1[j], y1[j], x2[j], y2[j]);
          }
        }
      }
    }
    t1 = 0;
    t2 = 1;
    for (i = 0; i < 4; i++)
    {
      temp = q[i] / p[i];
      if (p[i] < 0)
      {
        if (t1 <= temp)
          t1 = temp;
      }
      else
      {
        if (t2 > temp)
          t2 = temp;
      }
    }
    if (t1 < t2)
    {
      xx1 = x1[j] + t1 * p[1];
      xx2 = x1[j] + t2 * p[1];
      yy1 = y1[j] + t1 * p[3];
```

```
            yy2 = y1[j] + t2 * p[3];
            line(xx1, yy1, xx2, yy2);
        }
    }
    delay(5000);
    closegraph();
}
```

Output:



Note: Large values were taken for better visualization purpose.

Conclusion:

The Liang-Barsky Line Clipping Algorithm stands as a notable milestone in the field of computer graphics, offering a streamlined approach to line clipping with its emphasis on direct parameter calculations. Its efficiency and accuracy make it a valuable tool for rendering visible segments within a convex clipping window, contributing to the creation of visually compelling and realistic graphics. As technology continues to evolve, algorithms like Liang-Barsky remain a cornerstone in the pursuit of efficient and effective graphical rendering.