**Liang-Barsky Algorithm Report**

***Introduction***

Introduce the concept of line clipping and its importance in computer graphics and image processing. Briefly explain what the Liang-Barsky algorithm is and its primary purpose.

***Problem Statement***

Explain the problem of line clipping and the need for algorithms like Liang-Barsky to efficiently determine the portions of a line segment that are inside the clipping window.

***Liang-Barsky Algorithm Explanation***

Provide a detailed explanation of how the Liang-Barsky algorithm works. Break down the algorithm into steps:

**Parameter Calculation:**Explain how the algorithm calculates the parameters \(p\_1\), \(p\_2\), \(p\_3\), and \(p\_4\) based on the clipping window's coordinates and the line segment's endpoints.

**Line Classification:**Describe how the algorithm classifies the line segment into one of the categories: completely visible, completely invisible, or partially visible.

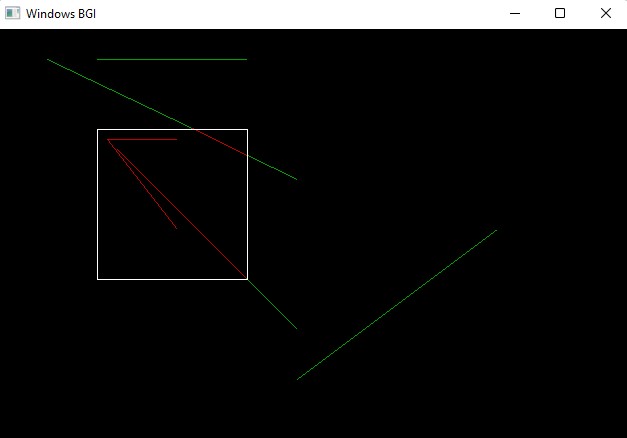
**Updating Endpoints:** Detail the process of updating the endpoints of the line segment to account for the clipping window boundaries, if necessary.

**Drawing Clipped Line:**Explain how the final clipped line segment is drawn if it's partially visible.

**Source Code:**

|  |  |
| --- | --- |
| #include<iostream>  #include<graphics.h>  #include<math.h>  #include<dos.h>  using namespace std;  void LiangAlgo(int x1, int y1, int x2, int y2)  {      int i;      Int xmin, xmax, ymin, ymax, xx1, xx2,yy1,yy2,dx,dy;      float t1,t2,p[4],q[4],temp;      xmin=100;      ymin=100;      xmax=250;      ymax=250;      dx=x2-x1;      dy=y2-y1;      p[0]=-dx;      p[1]=dx;      p[2]=-dy;      p[3]=dy;      q[0]=x1-xmin;      q[1]=xmax-x1;      q[2]=y1-ymin;      q[3]=ymax-y1;      for(i=0; i<4; i++)      {          if(p[i]==0)          {              if(q[i]>=0)              {                  if(i<2)          {              if(t1<=temp)                  t1=temp;          }          else          {              if(t2>temp)                  t2=temp;          }      }      if(t1<t2)      {          xx1 = x1 + t1 \* p[1];          xx2 = x1 + t2 \* p[1];          yy1 = y1 + t1 \* p[3];          yy2 = y1 + t2 \* p[3];          setcolor(2);          line(x1,y1,xx1,yy1);          setcolor(4);          line(xx1,yy1,xx2,yy2);          setcolor(2);          line(xx2,yy2,x2,y2);      }         else      {          setcolor(2);          line(x1,y1,x2,y2);      }  } | {                      if(y1<ymin)                      {                         y1=ymin;                      }                      if(y2>ymax)                      {                         y2=ymax;                      }                     setcolor(2);                   line(x1,y1,x2,y2);                  }                  if(i>1)                  {                      if(x1<xmin)                      {                         x1=xmin;                      }                      if(x2>xmax)                      {                         x2=xmax;                      }                     setcolor(2);                line(x1,y1,x2,y2);                  }              }          }      }      t1=0;      t2=1;      for(i=0; i<4; i++)      {          temp=q[i]/p[i];          if(p[i]<0)     int main()  {      int gd=DETECT,gm;      initgraph(&gd,&gm,"c:\\turboc3\\bgi");      int xmin,xmax,ymin,ymax;      xmin=100;      ymin=100;      xmax=250;      ymax=250;      LiangAlgo(120,120,300,300);      LiangAlgo(50,30,300,150);      LiangAlgo(50,30,300,30);      LiangAlgo(300,350,500,200);      LiangAlgo(110,110,180,200);      LiangAlgo(110,110,180,110);      setcolor(15);      rectangle(xmin,ymin,xmax,ymax);      delay(100000);      closegraph();  } |

***Output:***



***Applications***

Discuss the practical applications of the Liang-Barsky algorithm in computer graphics, image processing, and related fields. Provide examples of situations where this algorithm is particularly useful.

***Conclusion***

Summarize the key points discussed in the report and emphasize the significance of the Liang-Barsky algorithm in solving the line clipping problem.