BASIC PRINCIPLES OF COMPUTER USE ON THE VISUALIZATION OF EUCLIDEAN GEOMETRIC ENTITIES IN SPACE.

Author:Jorge Barros de Abreu
Email:ficmatin10@gmail.com

Abstract

Shows a way of using the computer as an auxiliary teaching tool in the visualisation of points, lines, planes, etc, through the free software Geomview and the free geometric routines library for computer programs called OOGL. The article is intended to elementary teachers and shows how to create basic geometric entities and software usage to view the results.

Keywords: spatial geometry computer geomview OOGL

1 Introduction

Geomview and OOGL were developed by the "Geometry Center" of University of Minnesota (USA). Despite the "Geometry Center" being discontinued, geomview kept going and it's an excellent tool for three-dimensional object viewing. All the methods of picture creation shown here can be reduced to the assembly of spatial shapes defining them as composed by polygons on the plane. For instance, the regular tetrahedron is composed of four equilateral triangles and the single-leaf straight cone can be viewed in the computer as being formed by 100 isosceles triangles and a base made of a 100-sided regular polygon. Otherwise, you can define a tetrahedron showing only the points of the four vertices, but also can show, besides of the points of the four vertices,four central points of each face or more three points in each face, or more hundred points in each face. In the computer your tetrahedron would show itself more defined each time you increase the number of points. It's worth noting that the geomview treats points as vertices and brings everything to the nearest unit cube but this can be modified.

2 Coordinate System

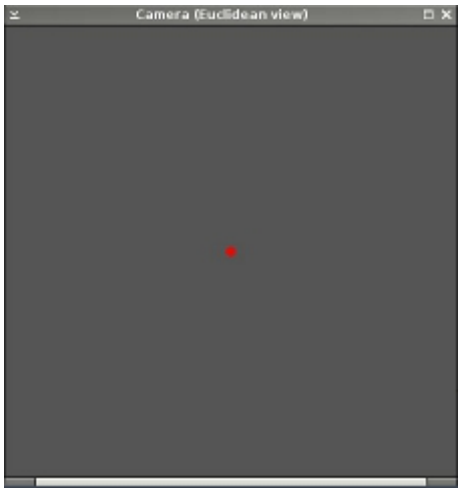
Geomview carries a file called axis.list that shows the coordinate axes in 3D euclidean space in the [-1,1] interval for the three axes.

3 Point in three-dimensional Space

To draw a point in 3D euclidean space, create the file point.vect:

```
(progn
   (new-geometry "[Point]"
      appearance {
          linewidth 10 #controls the diameter of the point
      }

      VECT
      1 1 1       # one polygonal line with a
              # vertex and one color.
      1           # one vertex in the polygonal line
                  # because we're creating a point.
      1           # one color.
      0 0 0       # the xyz coordinates of the point.
      1 0 0 1     # the red color of the point
   )
   (normalization "World" none)
)
```

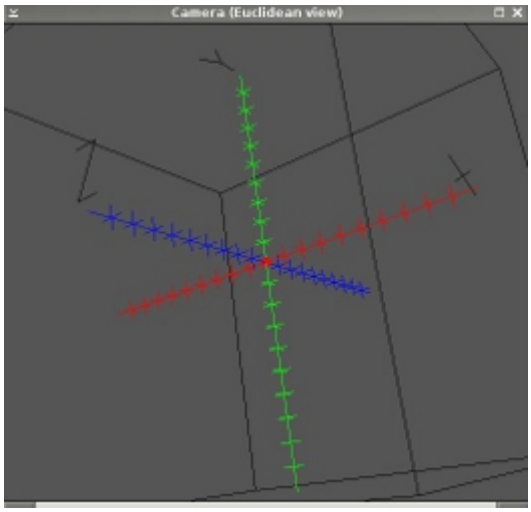


The command to obtain the above visualization is:

```
geomview -b 255 255 255 point.vect
```

3.1 Point in the Axes

To place the above point in the coordinate axes system, locate the file axis.list that comes with geomview, copy it in the same directory as ponto.vect.



The above image is obtained with:

```
geomview -b 255 255 255 ponto.vect axis.list
```

3.1.1 Three Points

Create the file called “three_points.vect” and put inside it the following content:

```
(progn
   (new-geometry "[Three Points]"
      appearance {
          linewidth 10 #controls the point diameter
      }

      VECT
      3 3 3       # in the whole file we have three polygonal
                  # lines with three vertices and three colors.
      1 1 1       # one vertex in each polygonal line because
                  # we are creating one point in each polygonal.
      1 1 1       # one color for each ``polygonal line'' that
                  # in this case is a point.
      0 0 0.3     # the xyz coordinates of point 1.
      0 0.3 0     # the xyz coordinates of point 2.
      0.3 0 0     # the xyz coordinates of point 3.
      1 0 0 1     # point 1 color
      0 1 0 1     # point 2 color
      0 0 1 1     # point 3 color
   )
   (normalization "World" none)
   (bbox-draw "[Three Points]" off)
)
```

3.1.2 Six Points

Create the file called “sixe_points.vect” and put inside it the following content:

```
(progn
      (new-geometry "[Six Points]"
      appearance {
          linewidth 10 #controls the point diameter
      }

      VECT
      6 6 6       # in the whole file we have six polygonal
                  # lines with six vertices and six colors.
      1 1 1 1 1 1 # how many vertices in each polygonal line.
      1 1 1 1 1 1 # one color for each ``polygonal line''
                  # in this case is a point.
      0 0 0.3     # the xyz coordinates of point 1.
      0 0.3 0     # the xyz coordinates of point 2.
      0.3 0 0     # the xyz coordinates of point 3.
      0 0 -0.3    # the xyz coordinates of point 4.
      0 -0.3 0    # the xyz coordinates of point 5.
      -0.3 0 0    # the xyz coordinates of point 6.
      1 0 0 1     # point 1 color
      0 1 0 1     # point 2 color
      0 0 1 1     # point 3 color
      1 1 0 1     # point 4 color
      1 1 0 1     # point 5 color
      0 1 1 1     # point 6 color
   )
   (normalization "World" none)
   (bbox-draw "[Six Points]" off)
)
```

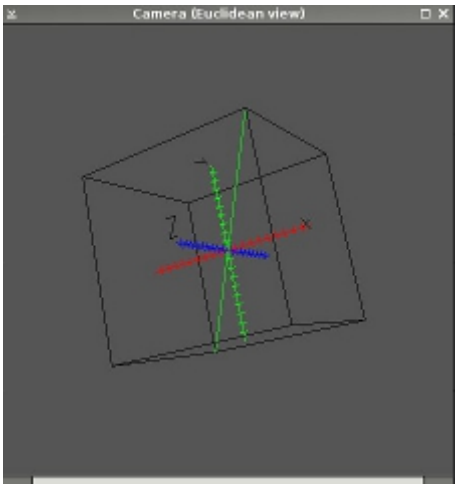
4 Segments in Space

To draw a segment that goes from (-1,-1,-1) to (1,1,1), create a file with the following content.

```
VECT
1 2 1         # one line with two vertices and one color.
2         # two end vertices of the segment
1         # one color
-1 -1 -1  1 1 1 # the beginning and end coordinates
0 1 0 1         # the color
```

Save the file with the label “segment.vect”.
To place the above segment in the axis system use:

```
geomview -b 255 255 255 segment.vect axis.list
```



In practice this way of drawing the segment is the same for drawing lines. Now enjoy with the following command that:

```
geomview -b 255 255 255 segment.vect three_points.vect axis.list
```

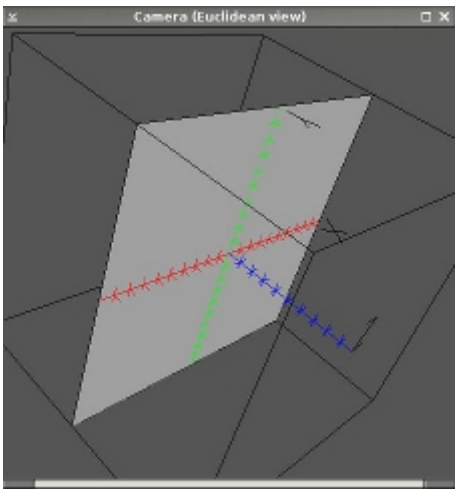
Again use the following command:

```
geomview -b 255 255 255 segment.vect six_points.vect axis.list
```

5 Planes

A plane can be represented by a four-sided polygon. Rigorously speaking we need only three points to define a plane, but for aesthetical reasons we'll use the four sides.

The picture below was obtained with "geomview square.quad axis.list" and can be used to represent the xy plane:



The file square.quad comes with geomview.

Now look at what happens with (type in one line on the terminal):

```
geomview -b 255 255 255 segment.vect six_points.vect
axis.list square.quad
```

Let's show the three coordinate planes xy, yz and xz. The command "geomview square.quad axis.list" shows the xy plane and the three-dimensional axis. We need to build the yz plane and we'll do it based on square.quad. Copy the file square.quad naming it xy.quad and again naming it yz.quad. Change the file yz.quad in a way so it's as the following:

```
QUAD
     0   -1   -1
     0   -1    1
     0    1    1
     0    1   -1
```

Create the file xz.quad:

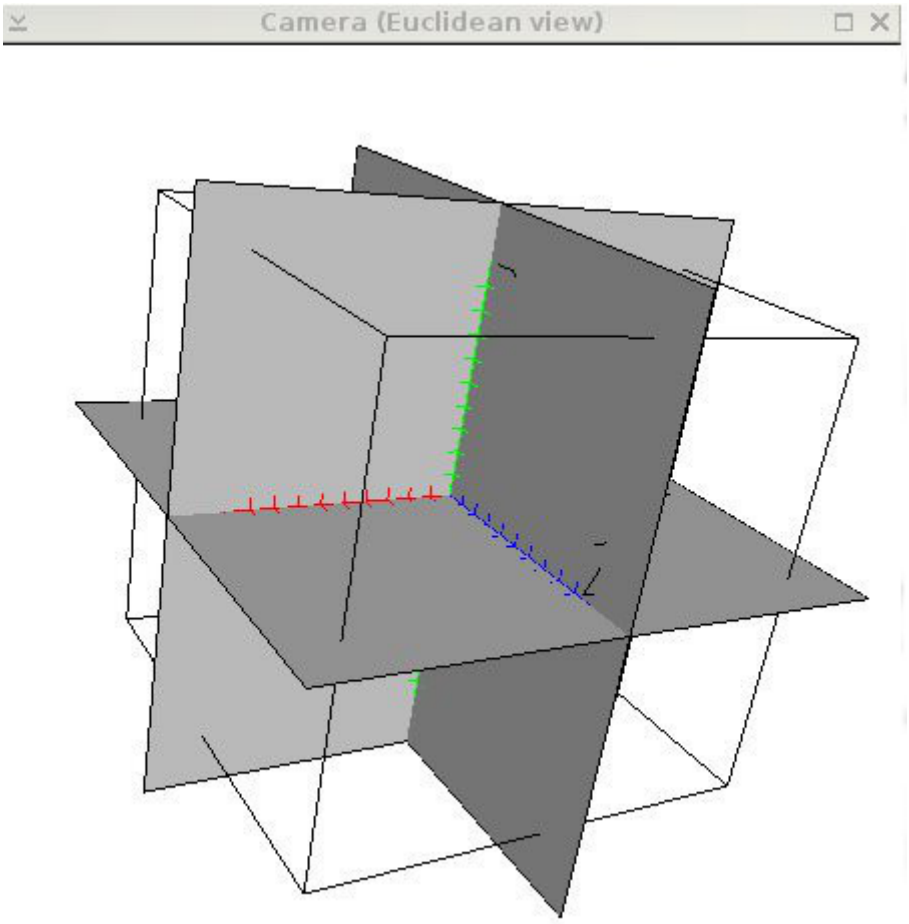
```
QUAD
     1    0    1
     1    0   -1
    -1    0   -1
    -1    0    1
```

The file xy.quad is like this:

```
QUAD
    -1   -1    0
     1   -1    0
     1    1    0
    -1    1    0
```

Now look at what happens with (type in one line on the terminal):

```
geomview -b 255 255 255 axis.list xy.quad yz.quad xz.quad
```



Drawing in geomview the plane that passes through the points (0,0,0), (-1,-1,-1) and forms a -45 degree angle with the plane xz. Save the file with the name 45_0.quad.

QUAD

-1 1 -1
-1 0 1
1 -1 1
1 0 -1

The general equation of the plane represented above is:

$$x+2y+z=0 \tag{1}$$

Drawing in geomview the plane that passes through the points (0,0,0), (-1,-1,-1) and is perpendicular to the previous plane. Save the file with the name 45_1.quad.

QUAD

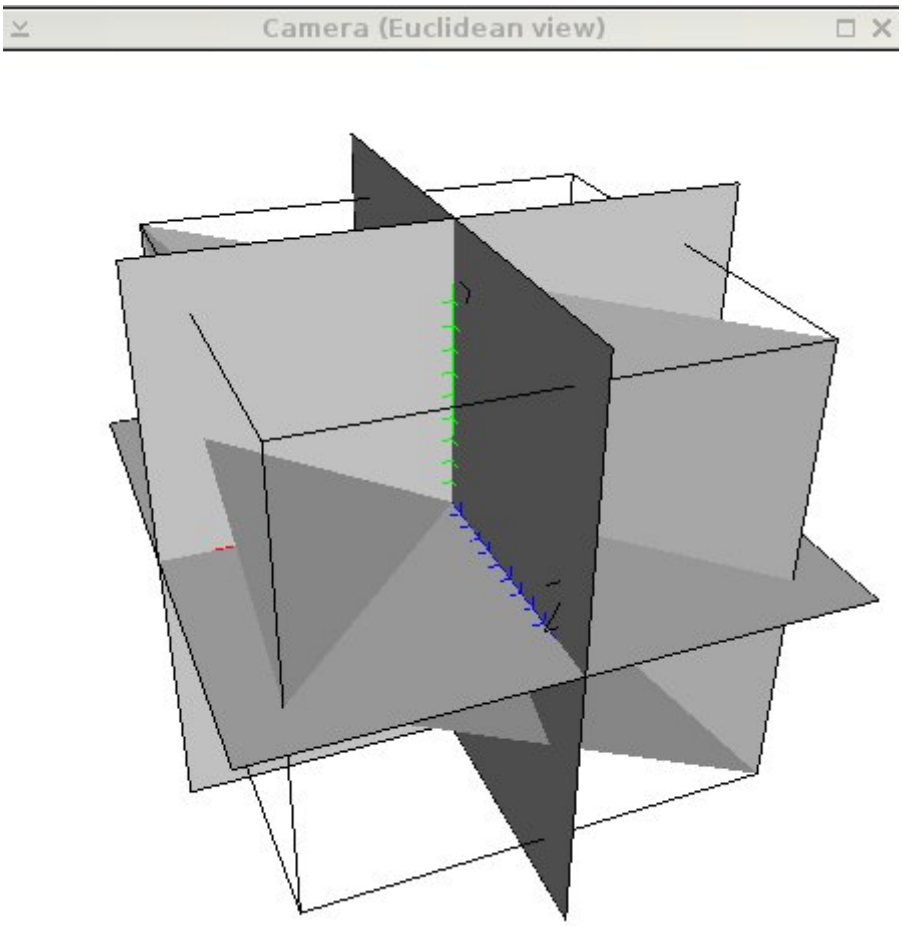
-1 1 -1
1 1 1
1 -1 1
-1 -1 -1

The general equation of the plane represented above is:

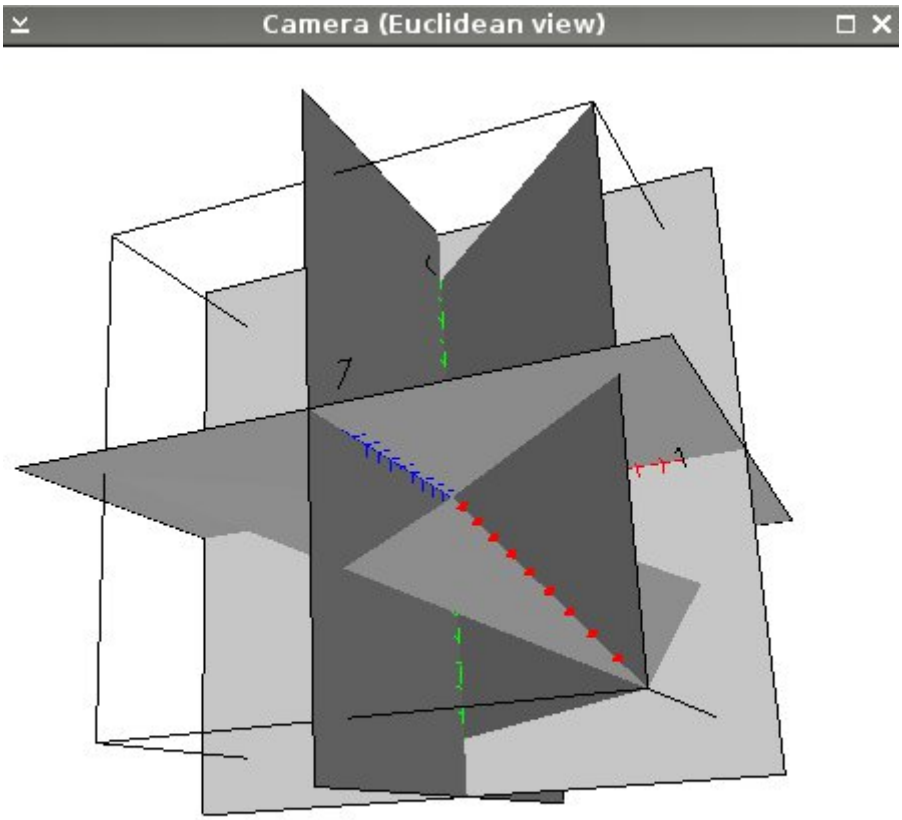
$$x-2y+z=0 \tag{2}$$

To visualize the two planes above the coordinate axis use:

```
geomview -b 255 255 255 axis.list 45_0.quad 45_1.quad
xy.quad yz.quad xz.quad
```



We have also something like this:



Note that the intersubsection between those two planes forms a line.

References

[Phillips(2008)] Phillips, M. et all. *Geomview Manual*. http://github.com/geomview/geomview/blob/master/doc/geomview.texi.