

SOFTWARE ASPECTS OF BUSINESS GRAPHICS†

WM. RANDOLPH FRANKLIN

Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy,
NY 12181, U.S.A.

(Received 24 November 1982)

Abstract—This paper considers some software and technical issues involved in the production of high quality three dimensional business graphics. It considers whether to use the vector or raster mode, the location or generation of source materials such as type fonts and geographic maps, existing 3-D plotting programs, and some technical issues in the calculation of 3-D plots. Finally it gives some examples of 3-D color plots illustrating different color techniques.

INTRODUCTION

Computers have been used to produce graphics for over three decades—Project Whirlwind at the Massachusetts Institute of Technology in 1950 had CRT's[7]. Interactive graphics input also dates from the 1950s when light pens were used by the United States SAGE (Semi-Automatic Ground Environment) Air Defense system. Graphics for those with only a line printer became widely available with the Harvard University Lab for Computer Graphics and Spatial Analysis SYMAP program in the 1960s[6]. However, because of the high cost, the low quality of the output, and the lack of color, until recently the major users of graphics have been academic researchers[12]. The major factors which have combined to change the situation in recent years are:

- (1) The falling cost of memory and storage: a megabyte now costs only 10,000–\$20,000, and Winchester drives with 30 megabytes of storage are common.
- (2) The availability of low cost color hard-copy units: an 8 by 10 instant color print or overhead transparency can be produced in 5 minutes for 5–\$8 on a camera costing about \$10,000. An instant color slide can be produced in a few minutes for about \$5.
- (3) The declining cost of computation: a mid-computer with 32-bit addressing capability that can multiply two floating point numbers in a few microseconds can be obtained for under \$200,000.

Thus the large, untapped market of business graphics has become established. This paper will consider some of the technical issues that arise in producing such plots. Some 3-D color examples will be given from the program COLORVIEW. We will not consider equally important issues of the esthetic design and choice of the right chart. Some excellent references of this are [13] which presents rules for the construction of effective charts, and [14] which shows how different aspects of a table of data can be highlighted by the choice of which functions of the items to plot.

We will study:

- (1) The choice of output format—vector or raster.
- (2) Source materials such as type fonts and geographic maps.
- (3) Existing three dimensional mapping programs.
- (4) Some technical problems in 3-D business graphics.
- (5) An example of a 3-D color business graphics program, COLORVIEW.

VECTOR VERSUS RASTER

Although raster graphics is displacing vector graphics, many high quality plots are produced on pen plotters. Some considerations are:

- (1) Fine quality hardcopy devices cost under \$10,000 in either case (vector eight pen plotters or color cameras).
- (2) Higher resolution is easier to obtain in the vector mode. A plotter resolution 1000 by 1000 is common, but obtaining that raster resolution requires a frame buffer with 1,000,000 pixels.
- (3) There are more colors available in the raster mode. Eight is the limit with a plotter unless the pens are changed during the plot, which requires that the vectors to be drawn be sorted by color. In the raster mode, allocating 24 bits per pixel allows more colors to be represented (16,777,216) than the human eye can distinguish.
- (4) The underlying algorithms used to produce the plot can be easier in one or the other modes. Here are 3 examples:
 - (i) Drawing a diagonal line between two points (x_1, y_1) and (x_2, y_2) , which is the fundamental operation of a pen plotter can be rather difficult on a raster device because simple methods such as the Bresenham algorithm can exhibit aliasing problems—the staircase effect. See [8] for a discussion of scan-conversion under conditions of limited memory, and [3] for a discussion of anti-aliasing.
 - (ii) Shading a polygon is a natural raster operation that is sometimes a hardware primitive, whereas a pen plotter must cross-hatch the polygon.
 - (iii) Overlaying text on the plot so that the grid lines behind it are deleted is simple with a raster frame

†This material is based in part upon work supported by the National Science Foundation under grant ECS 80-21504. The illustrations and use of COLORVIEW are courtesy of Hudson Data Systems, Inc.

buffer but requires a clipping operation in the vector mode.

In general, neither vector graphics nor raster graphics is clearly ahead of the other yet for producing business graphics.

SOURCE MATERIALS

Maps

If the business plot is to show geographic information, then we must obtain the map, i.e. the (x, y) coordinates of points along the borders of the regions, such as countries or states. There are assorted high resolution computer databases available at varying prices, but these are not necessary. The users of our plots will not be navigating with them, so we need only sufficient accuracy that the regions are recognizable. Using maps of lower resolution has several other benefits also:

- (1) The databases will be smaller and easier to store. The coterminous United States with 49 polygons can be represented with under 500 points.
- (2) Vector algorithms such as polygon filling that depend on the number of edges in each polygon will run faster, and the polygons with fewer edges will be faster to plot.
- (3) Some raster algorithms, such as 3-D shading break down when presented with polygons that are less than one pixel wide.
- (4) We can generate the maps ourselves with a data tablet.

If we generate the map ourselves, the best map data structure is chain-based. Here the map is a set of chains, with each chain being the border between two regions of the map and containing:

- (1) An ordered sequence $((x, y)_i, i = 1, N)$ of the points along the border.
- (2) The number of those points.
- (3) The names of the polygons on either side of the chain.

The map of the US mentioned above has about 160 chains. A simple Fortran program allows the user to take input from the data tablet, and annotate it. Finally the data can be cleaned up by using a text editor to make the endpoints of all chains meeting at a common point to be the same. This is the procedure using for the map of Europe in [9], and Fig. 4 here.

Even if a cartographic database is available, it can be quicker to follow the above procedure than to convert the database to the required format and delete unnecessary high-accuracy points. If no data tablet is available, then the map can even be entered by photocopying the map onto graph paper and then typing the coordinates in by hand. This is less onerous than it sounds and is further very simple to carry out; a map of New York State in the USA with 62 polygons and 1000 points was digitized this way by two people in three days, and with practice this could be reduced.

Nevertheless, if we wish to manipulate high resolution maps, there are many packages available, such as those listed in [10]. The raw data can be obtained

from sources such as the CIA's World Data Bank II[2]. This has international boundaries and state and provincial boundaries for the US and Canada, and world coastlines, islands, lakes, and rivers, for a total of 6,000,000 coordinate points and 30,000 line segments. A comparison of different levels of generalization (reduction in accuracy and number of points) of this data is given in [1].

Type fonts

The second source material that is needed is a set of type fonts with which to label the plot. If we are producing vector plots, then the standard is A.V. Hershey's set. The data, and a set of Fortran routines for manipulating it, are described in [16]. The Hershey set contains English and special characters in several fonts, such as Roman and Italic, in varying degrees of complexity. The more complex fonts are more realistic, but take longer to plot.

If we are producing raster plots, then we can scan convert the Hershey set, although a font designed from the start for raster display will look better. Raster fonts use a lot of storage: a single font of 100 characters, each 100 pixels high and wide, requires 128 K bytes to store, and a business graphics package should allow the user to select from several type styles and sizes. It is sufficient to store each style in its largest size, and then interpolate for the smaller sizes; however grey-level sampling should then be used[15]. This makes it possible for an upper and lowercase text string as small as five pixels high to be recognizable.

The concept of a text character can be generalized to include corporate logos and other small symbols. They can be entered into a raster database with a photo-digitizer, a TV camera, or if necessary by photocopying onto graph paper and typing strings of ones and zeros.

EXISTING THREE DIMENSIONAL PLOTTING PROGRAMS

There are many packages available for plotting functions of a variable over a geographic area. We will consider some three dimensional packages only here, since the two dimensional ones are so numerous and well known. First, Meusberger[11] at the University of Innsbruck can overlay columns of objects at points on the map. For example, a pile of coins can be used to represent the income of the region. The map is still drawn where it passes behind the piles, but this is not too noticeable if the piles are shaded.

David Douglas[5] at the University of Ottawa can plot pillars at specified locations on a map. For example, we can plot a map at the location of every city of over 100,000 population, with the height of the pillar proportional to the population. The map's outlines are correctly deleted when they pass behind the pillars.

Jim Dougenik[4] at Harvard describes a program, ASPEX, that can plot a continuous three dimensional

surface by a network of parallel lines. This is similar to the "fishnet" or "ruled surface" programs that have plotted functions $Z = F(X, Y)$ defined on a grid of points for over a decade.

SOME TECHNICAL PROBLEMS WITH THREE DIMENSIONAL BUSINESS GRAPHICS

There are many technical problems associated with three dimensional graphics. Some occur even when raster graphics is being used, in spite of the fact that an array of pixels seems to be a simple concept. Some common problems will now be considered:

(1) The finite precision of the computer's floating point arithmetic causes difficulties since although floating point numbers model the real number field, they actually satisfy very few of the real field axioms. For example:

(i) The associative law

$$(A + B) + C = A + (B + C)$$

doesn't hold even roughly if enough trailing bits are lost by normalization.

(ii) The distributive law

$$A*(B + C) = A*B + A*C$$

can fail for the same reason.

(iii) There may be missing reciprocals, i.e. no C for some A, B such that

$$A*B*C = A.$$

(iv) On at least one computer 1.0 is not a multiplicative identity, i.e. for some A

$$A*1.0 \neq A.$$

One effect of these is that if we start with several points in a plane, such as points defining the top of a prism, and then rotate, scale, and perspective transform them, they are no longer exactly in a plane. Thus if we are considering points to be hidden when they fall behind a plane, a plane may hide some of its own vertices. The obvious solution is to incorporate a fudge factor, i.e. to consider a point within a certain distance of a plane to be in front of it. This will fail for points which are really slightly behind a plane. Another solution is to know which planes a point is supposed to lie on, and then not check it against them.

(2) The fact that real numbers are discrete and not continuous can cause another problem if we start with a cube and rotate it, so that now the vertices are no longer exactly on the face planes. We might be satisfied with minor perturbations of the vertices so that they all fell exactly on some planes that were close approximations of the face planes. However, this might be impossible to achieve since each vertex must fall on three planes.

(3) If we start with an accurate map, of e.g. the

USA, and extend a prism side face up from each edge of the map, then extensive regions of the prism sides might be composed of side faces that are all each less than one pixel wide. This requires subpixel averaging, or else reduction of the input map so that there are no edges shorter than one pixel.

(4) The discretization of the pixels can cause problems. For instance, suppose that we are calculating the Z values of pixels along one edge of an oblique face, such as a prism top, in order to write it into a depth buffer (or Z -buffer). One common method, if the projected edge has a slope absolutely less than 45° , is to iterate along the edge in X . For each X , the naive solution is to find the corresponding Y and Z , round Y to the nearest integer, and set pixel $(X, \text{rnd}(Y))$. This will fail because $(X, \text{rnd}(Y), Z)$ does not fall exactly on the plane, and therefore two plane faces that actually abut along an edge will be interlaced in a zigzag dove-tail fashion in the resulting plot. What we must do instead is to substitute $(X, \text{rnd}(Y))$ into the plane equation to solve for Z .

(5) Another discretization problem similar to the above one can occur if we shade faces by scanning them horizontally, but draw lines by scanning them along the axis by which they vary most (i.e. in X if the line is mostly horizontal). This will cause a line that is adjacent to the edge of the face to sometimes occur slightly in front of the face and sometimes behind. These conditions will alternate along the line with a period of a few pixels, depending on the line's slope. One solution is to arbitrarily move all lines slightly closer to the viewer.

Some of the above problems are comparatively easy to solve, while others are expressions of fundamental complexities in the axiomatization of graphics. Lacking rigorous solutions, a good interactive debugger is useful here.

COLORVIEW EXAMPLES

We will now see some of the effects that are possible in business graphics by using color and three dimensions. They will be demonstrated using the program COLORVIEW, a proprietary product of Hudson Data Systems, Inc. COLORVIEW takes two inputs:

(1) A user defined base map of a geographic area that is divided into regions (a region can contain more than one disjoint polygon, as for example New York State being composed of the mainland and Long Island).

(2) A file specifying a value or height for each region.

COLORVIEW then erects a prism on each region of a height proportional to the value. Finally, it draws the visible surfaces with shading and colors in the desired format. COLORVIEW has over two dozen user settable parameters, controlling the placement of the map, colors, amount of shading, etc. The device and placement parameters are:

- (1) The name of the map file.
- (2) The name of the heights file.

- (3) The output device (the terminal for debugging, or the DeAnza for good copy).
- (4) The resolution of the plot.
- (5) The location of the plot on the screen.
- (6) The location of a surrounding box.
- (7) The height of the highest prism, as a fraction of the width of the plot.
- (8) The elevation angle for viewing the plot.
- (9) The azimuth angle for viewing the plot.

The shading parameters are:

- (1) The direction of the sunlight or assumed light source.

- (2) The ratio of specular light from the sun to overall diffuse lighting.

- (3) The amount by which the colors of farther faces should be desaturated to give an effect of distance.

The color parameters are:

- (1) The color of the background at the bottom of the plot.
- (2) The color of the background at the top (colors in between are linearly interpolated).
- (3) The color of the surrounding border.
- (4) The color of the edge lines of the prisms.
- (5) The color format for the prism sides.
- (6) The color of the prism sides if only a single color is desired.
- (7) The color format for the prism tops.
- (8) The color of the tops if only a single color is desired.
- (9) The name of the spectrum file for shading the sides and tops, if that format is being used.
- (10) Whether the prism side edges should be drawn, and if so whether all of them or just those "silhouette edges" where the prism side bends back on itself.
- (11) Whether the prism top edges should be drawn.

COLORVIEW has an associated program, SET-ITT, that allows the user to interactively define a *spectrum* or sequence of colors up the sides of the prisms. If this format is chosen, then the tallest prism will have the full spectrum running up its sides, a prism half as tall will have the lower half of the spectrum, and so on. There are a total of three different side shading formats:

- (1) Shade the prism sides as given above.
- (2) For any given prism, make all of its sides the same color, although different prisms may be different colors. Choose the color for each prism by indexing its height into the spectrum.
- (3) Make all the prisms' sides the same specified color.

There are two different shading formats for the prism tops:

- (1) Color each prism's top by indexing its height into the specified spectrum.
- (2) Make all the prism tops the same specified color.

The options to make all the sides or tops the same color can be used if the tops or sides, respectively, are

being colored differently to make the total plot more subdued and less garish.

The COLORVIEW algorithm is fundamentally quite simple; just an intensity buffer of size $512 \times 512 \times 3$ bytes storing the color of each pixel, and a depth buffer of 512×512 16-bit words storing the distance of the object visible in each pixel from the viewer or synthetic camera. The problem is difficult only because of the number of technical details, such as mentioned in the previous section, that must be addressed.

The COLORVIEW program is implemented in Ratfor on a Prime 750 computer with 2 MB of main memory. The program is about 2000 lines of code and uses about 1.5 MB of memory, mostly in the two large arrays. The images are generated on a DeAnza Imagearray processor with $512 \times 512 \times 24$ bits per pixel of image memory and 4 bits per pixel of overlays. However, COLORVIEW does not use the array processing capabilities of the DeAnza, and does all the calculations on the Prime. The images are displayed on a Conrac monitor. The copies are made with a Dunn camera exposing Polaroid Polacolor ER 8 by 10 inch Land film, type 809. We have also made copies onto 8 by 10 inch Polaroid overhead transparency film, and onto 35 mm. Ektachrome slide film. The execution times to calculate the figures shows here ranged from 45 to 64 CPU seconds per picture on a heavily loaded system.

Four examples of different techniques are now given. The first three are plots of New York State in the USA. This map has 62 regions representing counties. It is defined by 176 chains containing 747 points.

- (1) The first plot shows the number of licensed drivers in each county in 1980. The angle of view is 30° west of south, the elevation angle 45° , and the highest prism is 50% as high as the width of the plot. The sunlight comes from the west, and there is diffuse lighting which is as bright as sunlight falling normally on a face.

All the sides of each prism are the same color, depending on the height of that prism. The spectrum used was defined in a separate file to have four quarters of red, yellow, green, and blue, going from bottom to top. Thus any county whose number of licensed drivers is from 76 to 100% of the number of drivers in the highest county will have blue sides, any county with from 51 to 75% of the number will have green sides, and so on. The tops of the prisms are all the same color: cyan.

Some principles used in the display here are that of redundancy and selective display of data. Only one statistic is being shown, the number of drivers, and it is encoded in both the color and the prism heights. The heights provide a finer division, and allow the viewer to make some quick judgements about the distribution of the data at a glance. On the other hand, the colors allow prisms in different parts of the plot to be compared.

- (2) The second plot, also of New York State,

shows the number of convictions in each county for "driving while intoxicated" and "driving while ability impaired", i.e. for drunk driving. This shows the power of a plot over a table of numbers. We see that the counties of New York City, in the south-east, which have a large population, have few convictions.

The color format used here is to shade the sides of the prisms from bottom to top according to the spectrum file, and also to shade the tops according to the prisms' heights. Thus the plot looks as though it was cut from a layer cake or a sheet of plywood.

(3) The third plot shows the ratio of convictions per licensed driver for each county in New York State in 1980. The counties in red have at most 25% as many convictions per driver as the highest county. We can see graphically that adjacent counties can have quite different conviction rates. Here are the numbers for the two extreme counties:

County	Drivers	Convictions	Ratio
Clinton	40,923	695	1.70 %
Queens	953,933	341	0.037%

However, with just a table of numbers, these statistics would be diluted, and would not grab our attention.

The color scheme here is to make all the prism sides cyan, and color the tops according to the prism height and the spectrum file.

It would have been possible to combine all three plots into one by making the prisms' heights depend on one variable, the colors of the sides on the second, and the colors of the tops on the third. However, such a plot would be so cluttered that it would be harder to decipher than the original table of numbers.

This plot of conviction ratios also illustrates some potential pitfalls of computer graphics, since a plot is to a table of statistics as television news is to a detailed written report: it highlights a small subset of the data and ignores subtle details and footnotes. For example, since drivers are licensed by their county of residence, but convictions are reported in the county in which they occur, a tendency of drivers living in one county to get drunk in another, perhaps because of different regulations, will distort the figures. It is easy to report qualifications such as this below a table, but difficult to insert into a slide show.

Figures 3 and 4 illustrate another trade-off in choice of colors. Here the red, blue, and cyan are somewhat desaturated to make the colors less pure, but brighter. Color photography of business graphics is still an art since colors on the film can vary quite a bit from those shown on the CRT. For example, a yellow that looks somewhat dirty brownish on the CRT becomes quite brilliant on the film. A neutral grey can become blue-grey. Pure reds and blues appear much darker on the film. The problems arise since the CRT is displaying additive colors emitted by certain phosphors, but the film is reflecting sub-

tractive colors from different dyes that do not match the phosphors. Good color separation cameras that use a black and white CRT and a color wheel to expose the color film correct much of this, but not all.

(4) The final plot shows IEEE Computer Society members in each country of Europe. The shading format here is to make the sides and top of each prism one color. The color of each prism depends on its height, as indexed into the spectrum file.

SUMMARY

Computer graphics for business purposes is finally coming of age. It is possible to use both color and three dimensional plots. Data bases are available, and can also be created easily. The cost of hardware, both to compute and to display the plot, is now reasonable. Color plots can now be produced more economically by computer in quantity than by a human artist. The computer also allows special effects that are difficult or impossible to draw by hand.

REFERENCES

1. D. E. Anderson, J. L. Angel and A. J. Gorny, World data bank II. Content, structure and application. *An Advanced Study Symposium on Topological Data Structures for Geographic Information Systems*, 16-21 Oct. 1977, Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, MA 02138, U.S.A. (1977).
2. Central Intelligence Agency, *World Data Bank II, General User's Guide, and Tapes 1,2,3,4 & 5*, MIS # 's CIA/DF-77/001-005 tapes and PB 271-869. National Technical Information Service, United States Dept. of Commerce, Springfield, VA 22161 (Computer Products Office) (1977).
3. F. C. Crow, A comparison of anti-aliasing techniques. *IEEE Comput. Graphics Appls* 1, 40-48 (1981).
4. J. A. Dougenik, *ASPEX Program Summary Description in Harvard Library of Computer Graphics*, 1979 Mapping Collection: *Mapping Software and Cartographic Data Bases*. Harvard University Laboratory for Computer Graphics and Spatial Analysis (1979).
5. D. H. Douglas, The pillar mapping program: two common tasks of the thematic cartographer. In *Harvard Library of Computer Graphics*, 1979 Mapping Collection: *Mapping Software and Cartographic Data Bases*. Harvard University Laboratory for Computer Graphics and Spatial Analysis (1979).
6. E. E. Dudnik, *Symap—User's Reference Manual for Synagraphic Computer Mapping*. Dept. of Architecture, University of Illinois at Chicago Circle, Chicago, Illinois (1972).
7. J. D. Foley and A. van Dam, *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, New York (1982).
8. W. R. Franklin, Evaluation of algorithms to display vector plots on raster devices. *Computer Graphics and Image Processing* 11, 377-97 (1979).
9. W. R. Franklin, Program translates statistics into 3-D color map of Europe. *IEEE Comput. Graphics Appl.* 2, front cover and p. 4 (1982).
10. International Geographical Union, Commission on Geographical Data Sensing and Processing. *Computer Software for Spatial Data Handling; Volume 1: Full Geographic Information Systems, Volume 2: Data Manipulation Programs, Vol. 3: Cartography and Graphics*. International Geographical Union, 17 Kippewa Drive, Ottawa, Canada, K1S 3G3 (1980).
11. P. Meusburger, G. Fasching and J. Schonegger, THEMAT software specially programmed for smaller

- systems. In *Harvard Library of Computer Graphics*, 1979 *Mapping Collection: Mapping Software and Cartographic Data Bases*. Harvard University Laboratory for Computer Graphics and Spatial Analysis (1979).
12. W. G. Nisen and W. R. Franklin, The maturation of computer graphics. *ICP Interface Manufacturing Engng* 3, 5-11 (1978).
 13. A. Paller, K. Szoka and N. Nelson, *Choosing the Right Chart—A Comprehensive Guide for Computer Graphics Users*. Integrated Software Systems Corporation, San Diego, CA (1981).
 14. A. Vinberg and J. E. George, Computer graphics and the business executive—the new management team. *IEEE Comput. Graphics Appls* 1, (1981).
 15. J. E. Warnock, The display of characters using gray level sample arrays. *Comput. Graphics* 14, (SIGGRAPH'80 Conference Proceedings), 302-307 (1980).
 16. N. M. Wolcott, *Fortran IV Enhanced Character Graphics*. Computer Science and Technology, United States Dept. of Commerce, National Bureau of Standards, Washington DC (1978).



Fig. 1. Number of licensed drivers per county of New York State in 1980.

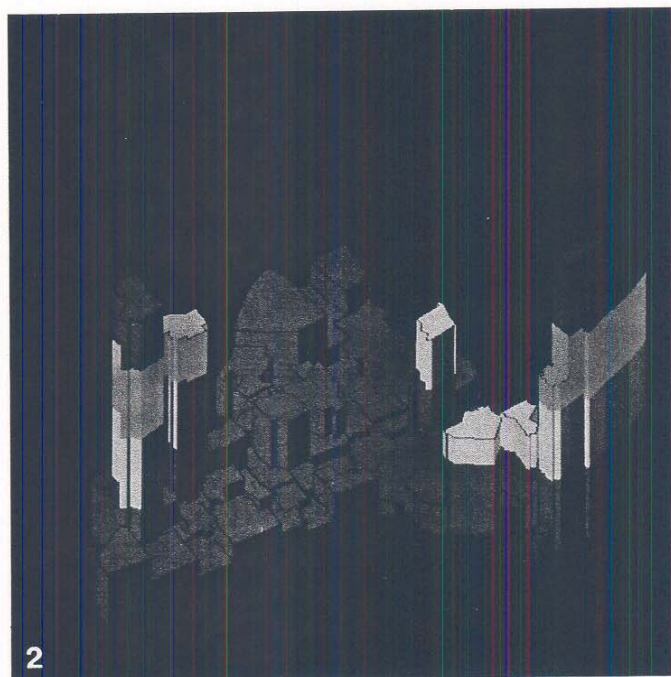


Fig. 2. Number of convictions for drunk driving per county of New York State in 1980.



Fig. 3. Ratio of convictions to drivers by county of New York State in 1980.

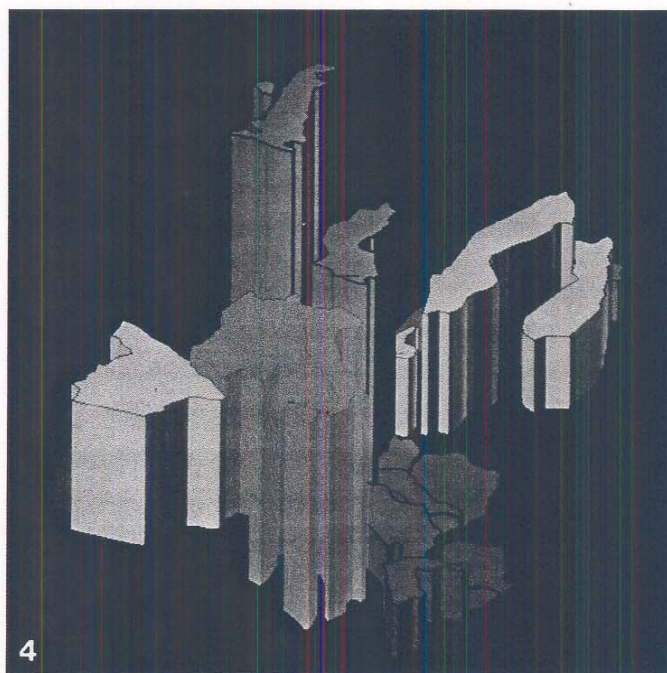


Fig. 4. Number of IEEE Computer Society Members per country of Europe.