VISUALIZING GEOGRAPHIC DATA THROUGH ANIMATION

Donna Okazaki
Department of Geography
University of Hawaii at Manoa
Honolulu, HI 96822

ABSTRACT

Visualization[*] of spatio-temporal data is enhanced through the use of various computer tools which make it possible to create realistically rendered, 3-D animations. This paper and video presentation will show the process and results of a cartographic animation project, which transformed paper maps and statistical tables into a presentation quality, 3-D animated map. Viewer reactions to the animation have been very positive, resulting in continuing effort in this area.

INTRODUCTION

Cartography is being continually impacted by advances in computing. This is particularly true in the case of geographic data visualization where technology has allowed data representations to evolve from static, printed maps to dynamic, interactive, graphic animations. High level software tools such as Khoros[**] (Rasure and Young 1992) allow cross-platform synthesis to take place, combining the strengths of different systems into a streamlined computational process. This makes it possible to create multiple perspectives of data which are vital to the scientific inquiry process.

The goal of this project was to create a thematic animation of Japan census data with the objective of being able to view and explore the urban development of Tokyo in time-space. Due to limitations of computer hardware and software, it was not possible to create an interactive, real-time, on-screen animation, so the results were transferred to videotape. However, it is important to stress the interactive nature of the animation process itself which employs visualization in many intermediate steps.

VISUALIZATION THROUGH ANIMATION

The main role of visualization in data analysis is to allow the researcher to interact with data representations during the investigative stage where data are explored to extract information or confirm results (Springmeyer, et. al. 1992). MacEachren and Ganter (1990) state that "The goal of cartographic visualization ... is to produce scientific insights by facilitating the identification of patterns, relationships, and anomalies in data." Animation is viewed as an effective tool

[*] In this paper, the term "visualization" is used exclusively in the context of research methods as a means of data exploration.

[**] An open software product available through anonymous ftp from ftp.eece.unm.edu or by requesting an order form via email to khoros-request@chama.eece.unm.edu.

for visualizing complex phenomena which can be mapped into time-dependent 2-D or 3-D data representations (DiBiase, et. al. 1992; Gershon 1992; Sparr, et. al. 1993). However, the technology to support animation poses challenges which are related to the availability of fast computer hardware, the resolution supported by video output devices and the state of currently available software.

The first challenge, inadequate computing power, is a common one. When available processor speed is insufficient to produce real-time, animated computer displays, videotaped animations are a cost-effective means of creating these unique data views. Animations of large amounts of data which result from complex computations are especially amenable to this low-cost approach (Choudry 1992). This allows visualization of numerically intensive simulation models or in the case of this project, allows the visualization of complexly rendered 3-D views of data.

The human visual system compensates for the second challenge. There is evidence to support the idea that animation enhances the visibility of features embedded in data. Gershon (1992) claims, "It has been found that observing motion triggers a set of fast and effective mechanisms in the visual system that allow the efficient detection of moving features." This would compensate for the loss of resolution resulting from transferring high quality computer images to consumer-grade videotape.

The third challenge is the most difficult. General purpose visualization packages such as Khoros (Khoral Research, Inc.), AVS (AVS, Inc.) and Explorer (SGI, Inc.) are sufficient for data viewing up to the computational limits of the system, but do not have many of the features of a complete animation system. They also aren't designed to capture and store geographic information. However, these packages are extensible, so they can be used to bridge between GIS and animation applications by importing data in one format, applying the appropriate translation and exporting them in another format. During the transformation process, as data move from package to package, the researcher gains insight from viewing the different representations of that data.

THE PRODUCTION PROCESS

The production of an animated, 3-D map of Japan census data involved the use of several programs. These were selected based on availability and ability to address the individual components of the cartographic process. The animation process was broken into five stages:

1. Capture and storage of the x-y spatial component, in this case the city, town and village boundaries of Tokyo and four neighboring prefectures. The PC-ArcInfo package was used to digitize and format this data.
2. Addition of thematic variables as the z spatial component and creation of texture maps* for overlay on the resulting 3-D surface. Khoros, on a Sun workstation, provided a visual programming environment conducive to this data transformation process.

* In the context of animation, texture map refers to any image which is draped over an object, such as a 3-D surface, giving it color and/or the appearance of texture.

3. Creation and rendering of individual video frames. Wavefront, a powerful animation package on a Silicon Graphics workstation, was used for this stage.
4. Final editing and composition. Macintosh and PC programs were used to create text overlays, backgrounds and other frames for the final editing. Khoros was used to assemble and preview the final video frame images.
5. Transferring final frames to videotape. The final frames were transferred over the campus network to a Diaquest animation controller which buffered and wrote each image to a frame accurate S-VHS recorder.

No single software package was sufficient to perform all these tasks. A combination of tools, with Khoros as the backbone, enabled the transition from paper maps and statistical tables to video tape.

<u>Khoros: Production Backbone</u>

Khoros is an extensible, visual programming system which is useful in a number of ways. It is a powerful data transformation tool as well as data visualizer, capable of creating different views of data and bridging between other, non-compatible software applications. The visual programming environment is flow-based, comprised of a workspace which is populated by input module icons linked to data processing module icons linked to output module icons. Figure 1 shows a workspace for viewing the population density

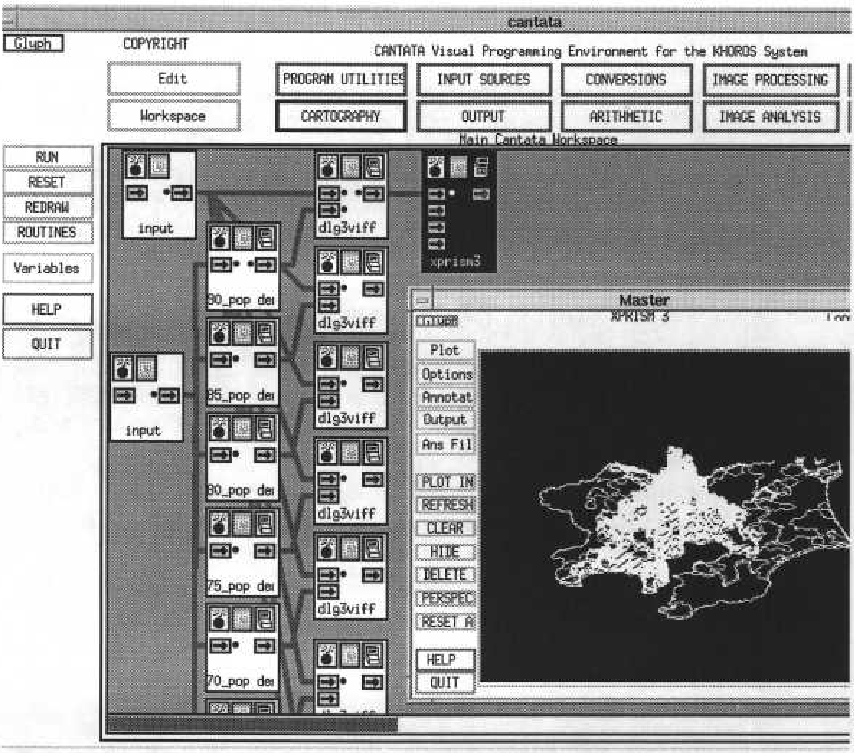


Figure 1. Khoros visual programming environment.

of cities, towns and villages for several census years. Tabular census data are merged with digitized boundaries and fed to a 3-D viewer which is shown displaying the data for 1990 as 3-D contours.

Extending the functionality of Khoros beyond its supplied modules requires C-language programming, but this is greatly assisted by interactive code generators. Two new modules were programmed for this project to input data from PC-ArcInfo and output to Wavefront object files. All other data transformation functions were provided by Khoros in its visual programming environment.

Stage 1: Digitizing

The city, town and village boundary maps for each prefecture were taken from a government data book (Jichisho Gyoseikyoku Shinkoka 1991). These were not ideal for digitizing and assembly since each was drawn at a different scale and did not precisely fit with adjacent prefectures. In addition, they did not contain any coordinate or projection information. This necessitated the use of a GIS, in this case PC-ArcInfo, which could rescale and fit these maps to a single geographic coordinate system. This was accomplished by first digitizing a base map of Japan prefectures[*] and transforming the digitizer coordinates to projected coordinates.[**] Then each city, town and village map was digitized and its coordinates transformed to fit the base map. The transformed data were then merged, prefecture boundaries were matched and polygon topology was built into a single coverage for the five prefectures centered around Tokyo city.

The boundary data were exported as a DLG3-Optional (U.S. Geological Survey 1990) formatted file. This format was used because it is supported by Khoros and its major and minor attribute keys fit perfectly with the prefecture and city-town-village ID's.

Stage 2: Assembling Key Data Objects

Census data for each city, town and village area, for the years 1955 through 1990, were entered into a spreadsheet on the Sun workstation. Boundary changes were handled by aggregation or by averaging to normalize the data to the 1991 boundaries. These calculations were handled by the spreadsheet software which also output the ASCII lookup tables which fed into the first Khoros module.

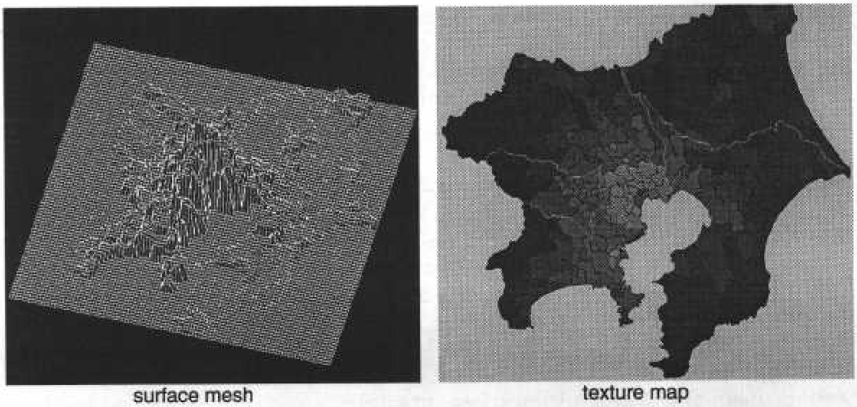
Khoros provides a module to read DLG3 files, but it is deficient in several respects, so a new module was programmed. This new module reads a DLG3-O file and optionally calculates the value of a key from each object's major and minor attribute codes, then uses this key to find a feature value in the lookup table. The data are output as a raster grid with either key values or attribute values written into each cell.

[*] Using a 1:200,000 scale map from the 1990 National Atlas of Japan published by the Japan Geographical Survey Institute.

[**] PC-ArcInfo does not support the map's oblique-conic projection, so the oblique-mercator projection was substituted. The introduced error was not considered significant for this project at this scale.

Several raster grids were created to produce both x,y,z surface objects and texture maps for input into the Wavefront system. One surface object was created per census year and, due to limitations in the Wavefront system, was limited to a 1 km cell resolution. A second Khoros module was programmed to take the grid and convert it into a rectangular mesh in a Wavefront object format file. Texture maps were created for each census year to color-code areas by population density. Khoros was then used to generate one texture map per video frame by morphing (interpolating) each census year's texture map into the next. These data files were exported to the Wavefront system which rendered the raw animation frames.

Figure 2 shows the surface mesh for 1990 and its associated texture map which has been converted to greyscale for this illustration.



surface mesh

texture map

Figure 2. 1990 population density as a surface mesh and texture map

Stage 3: Rendering of Animation Frames

The Wavefront system consists of several modules accessed by a common user interface. The model module was used to link a dummy texture map file to each surface object. Then the preview module was used to create and interpolate between animation key frames. This allowed a combination of object morphing and camera movement. Surface meshes were morphed over time from the 1955 through the 1990 population density surfaces. The camera was kept still during the forward surface morph and moved to another viewpoint while "rewinding" the data.

While in the preview module, it was possible to visualize a rough representation of the final animation. This was useful in determining the best orientation of the camera relative to the 3-D object. Several camera angles, over the same 35 year period, were picked which seemed to show the best views of urban growth from the city center. The first shows southward growth toward Yokohama, the second shows westward growth along a major transportation corridor and the third shows more recent growth in the northeasterly direction.

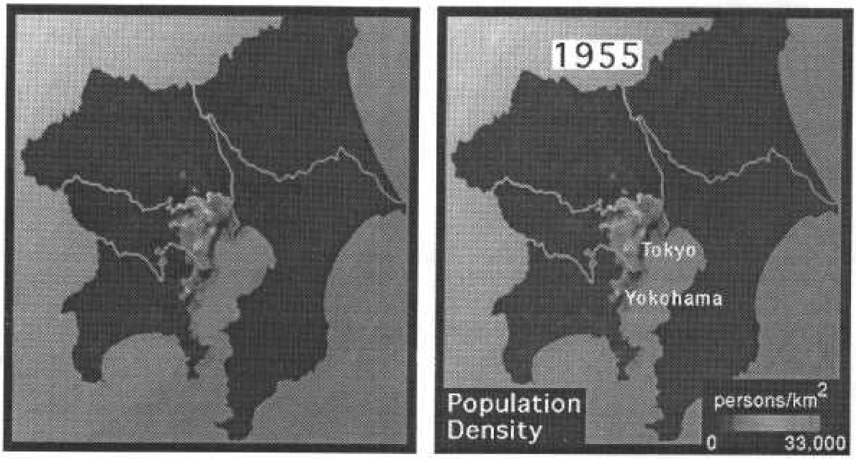
The creation of the video frames was accomplished by submitting batch process jobs to run over a period of several days. The batch job script substituted the correct Khoros generated texture map for the dummy file before rendering each frame. The output consisted of one raster image file per video frame (30 frames per second of viewing time). These raw, unenhanced frames were written directly to a frame accurate video recorder as a preview step to determine what changes should be made to the animation sequence.

For efficiency, the entire animation was broken into scenes which formed separate sequences of frames. As each scene was completed, its raw frames were transferred back to Khoros for final editing.

Stage 4: Editing and Composing

Introductory and ending title slides were composed and rendered using 3D-Studio on a PC and output in the same file format as the Wavefront rendered frames. Adobe Photoshop on the Macintosh was used to assemble a Landsat image of the data area which had been scanned in four segments. This same software was also used to create pieces of annotation text for pasting onto the animation frames.

Standard Khoros modules were used to interactively paste annotation into sample animation frames as illustrated in Figure 3. Once the layout was determined, these modules were run in a batch job to modify all necessary frames (a process which took several hours per scene).



raw video frame

annotated video frame

Figure 3. Video frame annotation

Other functions performed by Khoros included dissolving between title slides and dissolving between the introductory Landsat scene and the first data scene. Simple text slides were also assembled by pasting pieces of Photoshop text onto a blank background.

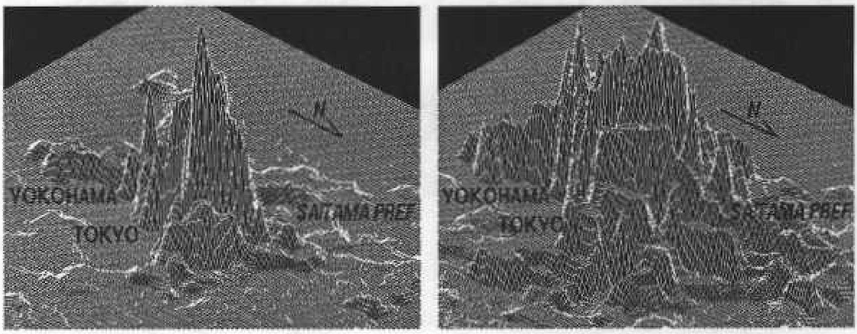
Stage 5: Videotaping

The last stage was the easiest due to the availability of high-end video production equipment. Transferring the animation frame images to S-VHS tape was done with a batch script on the Sun workstation. It sent recording commands and images across the campus network to a Diaquest animation controller which buffered and transferred each image to a frame accurate recorder. Frame accuracy meant that the animation could be transferred in pieces and written to precise locations on tape, a useful feature for making small adjustments. With a transfer rate of just 2 to 3 frames per minute, this step also took many hours per scene.

RESULTS

The final animation, which is less than three minutes long, clearly shows Tokyo city following a classic, doughnut-shaped pattern of growth. The overall pattern reveals an enlarging central core with low population density, surrounded by a widening, high density, suburban ring. However, it is evident that there are sub-patterns within that growth which lead to questions about transportation networks, local area urban planning behavior, land pricing and other factors which influence population growth. For example, the area between Tokyo city and nearby Yokohama city shows relatively higher growth than the area just northeast of Tokyo. Higher growth rate patterns can also be seen moving west and northwest from the city center.

In addition to being able to detect overall trends, animation makes it easier, over static maps, to detect anomalies or outliers which should lead to questions about the area or about the data. For example, in Figure 4, it is difficult to see the growth pattern of cities and towns in the southern region of Saitama Prefecture, north of Tokyo.



1955 Population Density

1990 Population Density

Figure 4. City and town growth north of Tokyo

The animation shows which cities and towns in Saitama appear to be small urban growth areas on their own, as opposed to overflow from Tokyo city. These small urban areas seem to grow as independent peaks, sometimes merging with higher density areas spreading out from Tokyo city.

Adding more dimensionality to the animation will make it possible to visualize the interaction of several variables. For example, using a texture map color-coded by land price or any other variable may reveal relationships between it and the surface variable, population density. Overlaying vector data such as transportation networks and distance indicators would also increase the information content of the display, thereby increasing opportunities for detecting trends and anomalies.

CONCLUSION

Interactive visualization tools are as important to the animation cartographer as they are to the researcher studying the area being mapped. The software packages described in this paper are part of a growing body of enabling technologies which assist us in developing clearer views of the world.

REFERENCES

Choudry, S.I., 1992, A Low Cost Approach to Animated Flow Visualization Using High Quality Video. *Advances in Scientific Visualization,* Berlin, Springer-Verlag, pp. 131-144.

DiBiase, D., A.M. MacEachren, J.B. Krygier, and C. Reeves, 1992. Animation and the Role of Map Design in Scientific Visualization. *Cartography and Geographic Information Systems* 19(4): 201-214, 265-266.

Gershon, N.D., 1992. Visualization of Fuzzy Data Using Generalized Animation. *Proceedings of Visualization '92,* pp. 268-273.

Jichisho Gyoseikyoku Shinkoka, 1991. *Zenkoku shichoson yoran,* Tokyo, Daiichi Hoki.

MacEachren, A.M., and J. Ganter, 1990. A Pattern Identification Approach to Cartographic Visualization. *Cartographica* 27(2): 64-81.

Rasure, J., and M. Young, 1992. An Open Environment for Image Processing and Software Development. *1992 SPIE/IS&T Symposium on Electronic Imaging, SPIE Proceedings* 1659: 300-310.

Sparr, T.M., R.D. Bergeron, and L.D. Meeker, 1993. A Visualization-Based Model for a Scientific Database System. *Focus on Scientific Visualization,* Berlin, Springer-Verlag, pp. 103-121.

Springmeyer, R.R., M.M. Blattner, and N.L. Max, 1992. A Characterization of the Scientific Data Analysis Process. *Proceedings of Visualization '92,* pp. 235-242.

U. S. Geological Survey, 1990. Digital Line Graphs from 1:24,000-Scale Maps: Data Users Guide 1, Reston, U.S. Geological Survey.