

A Collection of Visual Thesauri for Browsing Large Collections of Geographic Images[†]

Marshall C. Ramsey, Hsinchun Chen, and Bin Zhu

MIS Department, Karl Eller Graduate School of Management, University of Arizona, Tucson, AZ 85721.

E-mail: mramsey@bpa.arizona.edu

Bruce R. Schatz

Digital Library Research Program, University of Illinois at Urbana-Champaign, Urbana, IL 61801.

E-mail: schatz@csl.ncsa.uiuc.edu

Digital libraries of geo-spatial multimedia content are currently deficient in providing fuzzy, concept-based retrieval mechanisms to users. The main challenge is that indexing and thesaurus creation are extremely labor-intensive processes for text documents and especially for images. Recently, 800,000 declassified satellite photographs were made available by the United States Geological Survey. Additionally, millions of satellite and aerial photographs are archived in national and local map libraries. Such enormous collections make human indexing and thesaurus generation methods impossible to utilize. In this article we propose a scalable method to automatically generate visual thesauri of large collections of geo-spatial media using fuzzy, unsupervised machine-learning techniques.

Traditional thesaurus creation is an extremely labor-intensive process for text documents and especially for images. Currently, the University of California, Santa Barbara Maps and Image Library has more than two million aerial photographs of current and historical significance (Manjunath & Ma, 1996). Such enormous collections make human indexing and thesaurus generation slow and costly. According to Rorvig (1990) human indexing typically only provides sparse image descriptions because of the high cost. This leaves users with inadequate resources to efficiently search for images.

Besides volume, geographical images present other challenges to indexers. For example, images may be easily indexed by geographical location and time; however, content descriptions and semantics cannot. According to Carlson (1997) an image may contain a high density of information that subjective index categories cannot fully de-

scribe. Additionally, the content that is relevant to any individual is impossible to predict.

Recent work by researchers at the Alexandria Digital Library have provided a means to automatically analyze digitized photograph textures using Gabor filters (Gabor, 1946; Manjunath & Ma, 1996). The tags created by the filters can be combined with other location data such as the Advanced Very-High Resolution Radiometer (AVHRR) thematic data and other multispectral signatures taken via satellite. The assimilated data can be used as input to an unsupervised machine learning system to produce a browsable image thesaurus. Unlike the system developed by Rorvig, Turner, & Moncada (1999), ours does not leverage text descriptions of images. Instead, the search space consists solely of numeric representation of image textures and the interface consists of a two-dimensional map of significant textures extracted from the collection. Our system is most applicable to large collections of geographic images with sparse or no text descriptions.

In this article we propose a scalable method to automatically generate visual thesauri of aerial photographs and AVHRR data using Kohonen self-organizing maps (SOM). A detailed description of the texture-based thesauri creation is presented, followed by a discussion of the interfaces. Then, the AVHRR thesaurus is discussed and the article concludes with summaries of our findings and future work.

Texture-Based Visual Thesaurus Creation

The first step in constructing a visual thesaurus is to reduce the dimensionality of the visual scene which has already been lessened by the limits of film. The original scene has infinite resolution and very high spectral bandwidth and a photograph is limited in both respects. The digitization of the film further reduces the image dimensionality by imposing discrete color and typically lower resolution.

[†] The figures featured in this article can be viewed in color online, at <http://www.interscience.wiley.com/jpages/0002-8231/suppmat/index.html>.

The digitized image is a matrix of pixels that has lost much of the original scene information but still retains a significant amount. A total of 4,194,304 unique identifiable textures for a 128×128 pixel 256 grayscale image subset can be stored in this format. However, textures can have completely different pixel values and still be perceived as similar (Picard & Kabir, 1993).

A lossy classification system can be used to identify texture concepts rather than unique textures. The problem is finding the best discriminators for identifying conceptually (visually) similar textures. Commonly, texture information is reduced to small-feature vectors that are formed by summarizing the pixel information into statistical elements that can be used to discriminate a reasonable number of texture concepts. Each element in the feature vector represents a potentially useful discriminator of texture; however, the elements and value ranges that best determine texture concept are still unknown. A method must be devised to further summarize tile information into a few statically significant classifications by analyzing the feature vectors.

Step 1: Create Image Tiles and Reduced Resolution Images

Typically the aerial photos used in our study are 5000×5000 pixel grayscale images. Before analyzing their textures, the images must be split into small tiles that can be more easily analyzed. A tile is the fundamental unit for our processes and is a 128×128 pixel subset of an image. Extracting the tiles is trivial and involves reading the column-row order pixel vector of the original image in sun raster format and translating it to a concatenated list of column-row order raw format tiles. The resultant file is about twice the size of the original because each column and row of tiles overlap the previous one by 50%. The overlap is necessary to more accurately define region boundaries during segmentation.

Because the large aerial photos are cumbersome to view, requiring large amounts of computer memory to display, reduced-resolution versions are created for quick browsing. The down-sampled images are only used for display and are not processed during thesaurus creation.

On Silicon Graphics, Inc.'s (SGI) new Origin 2000 multiprocessor computer at the National Computational Science Alliance (NCSA), 28 images were simultaneously processed on 28 processors in approximately 10 minutes with an average of almost three images per minute.

Step 2: Extract Features

The "tile files" created in Step 1 are used to produce feature vectors of 60 floating point numbers. The numbers are generated by applying a bank of Gabor filters that analyze the texture of a tile. Gabor filters are scale tunable edge and line (bar) detectors which create statistics, describing the microfeatures, that are used to characterize the texture of a given region (Manjunath & Ma, 1996). The

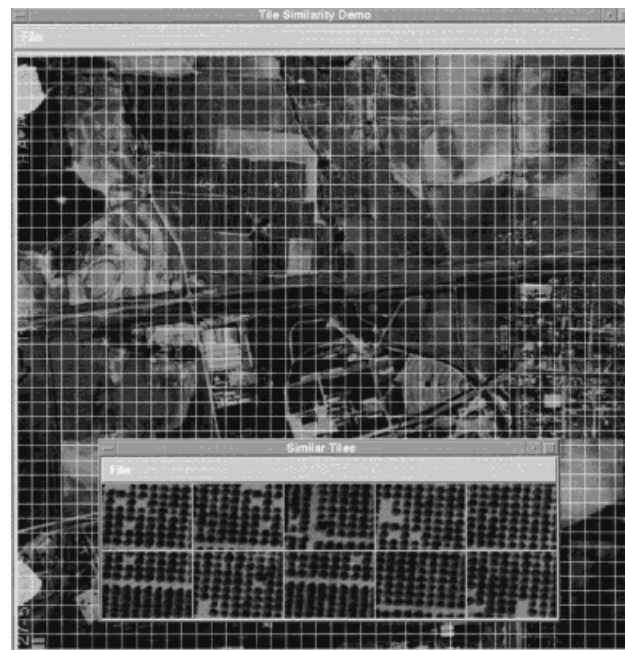


FIG. 1. Tile similarity program display.

filters are applied in six different orientations and five different scales. For each orientation-scale combination, a mean and standard deviation is calculated, thus, creating a total of 60 values. The feature vector of each tile is saved in the image's "feature file."

Although many different algorithms have been proposed for texture analysis, Gabor filters were chosen because, for both region similarity measures and comparison efficiency, they have been proven as good if not better than the common pyramid-structured wavelet transform, tree-structured wavelet transform, and multiresolution simultaneous autoregressive model (Manjunath & Ma, 1996).

In the remaining steps, tile similarity must be measurable and accurate. Ultimately, two tiles should be considered similar if they are visually alike. However, this is a subjective evaluation and is not quantifiable. Dissimilarity, and conversely similarity, can be calculated by using a distance metric between two feature vectors. Euclidean distance is the most common measure of dissimilarity and is used by our system. We claim that two tiles are similar if the Euclidean distance of their feature vectors is relatively small.

Before creating the visual thesaurus, we produced a system that finds similarly textured tiles in a single image. The program is used to visually verify that feature extraction has meaningfully captured the texture content and that Euclidean distance is an acceptable metric for computing dissimilarity. Figure 1 shows the program's display which starts with an image that has been segmented into tiles and analyzed using Gabor filters. When a tile in the image is selected a precomputed index file of the ten closest tiles, based on the Euclidean distance of feature vectors, is used to find and show similar tiles in a separate window. Our

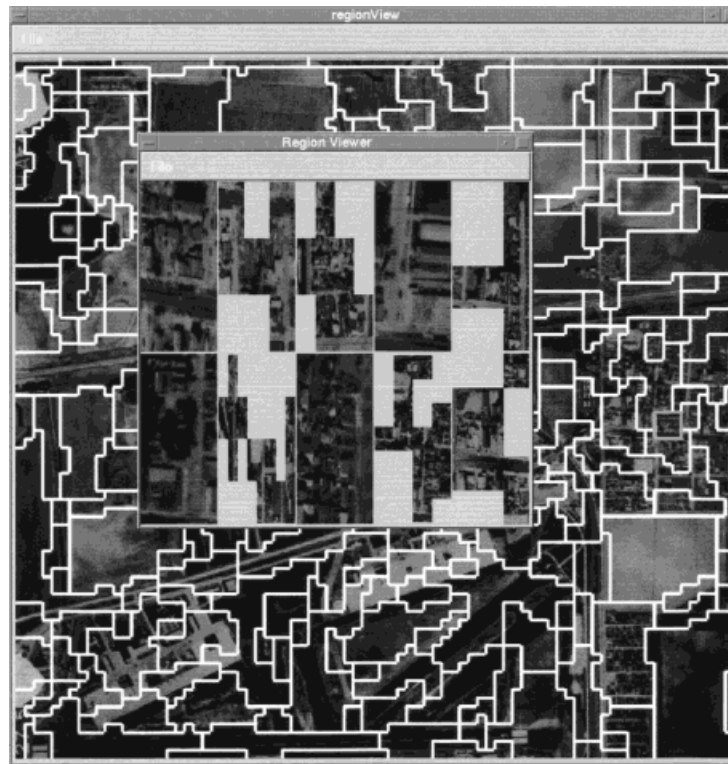


FIG. 2. Region similarity program display.

initial finding is that the closest tiles are visually similar to the associated tile.

Step 3: Segment Images Using Texture Flow Analysis

The previously created tiles and their feature vectors are used to segment the aerial photograph by grouping adjacent tiles with similar textures. Each tile's feature vector is compared with its eight neighbors and texture gradients are calculated. The gradients of similarly oriented tiles are combined to create a texture flow with direction and energy. The local texture flows are used to find region boundaries by identifying the areas with opposite orientations. This leads to a discontinuous set of boundary edges that are then connected to form regions. The number of regions are then reduced by merging similar regions that are adjacent (Ma & Manjunath, 1996).

Step 4: Find Representative Feature Vectors

After the images are segmented, a representative feature vector for the region's textures is calculated and saved to a "region feature file." We use the average feature vector for the tiles in a region because it should be a close approximation of the tile cluster's centroid in the multidimensional space.

Figure 2 shows the display of a demonstration program used to verify both the representative feature vectors and the Euclidean distance as a measure of dissimilarity. This ap-

plication is similar to the one used in Step 2. The demo displays an image segmented into regions that the user can select with the mouse. An index file of the closest regions is used to find and display the ten most similar regions. This index file is used only in the similarity demonstration and is not utilized in the visual thesaurus. The program visually demonstrates that average feature vectors and Euclidean distance can be used to determine region similarity.

Step 5: Create Self-Organizing Map

The algorithm used by the similarity demonstration to find the closest tiles has a computational complexity of $O(n^2)$, where n is the number of feature vectors. Because the algorithm increases exponentially, it is not scalable to large collections. A more efficient approach is to cluster textures and then let the user browse the clusters. Our experience with Kohonen SOM borrows from Lin's work (1997) and has shown them to be effective for the creation of scalable visual thesauri of texts (Chen, Schuffels, & Orwig, 1996). This algorithm has a time complexity of $O(n)$, where n is the number of feature vectors. Since input to input comparisons are avoided, the algorithm is only linearly increasing and, thus, scalable. For this reason we propose to extend the SOM to cluster and display textures of images.

An SOM is an unsupervised single-layer neural network used for dimensionality reduction and clustering. An advantage SOMs have over other clustering algorithms is the

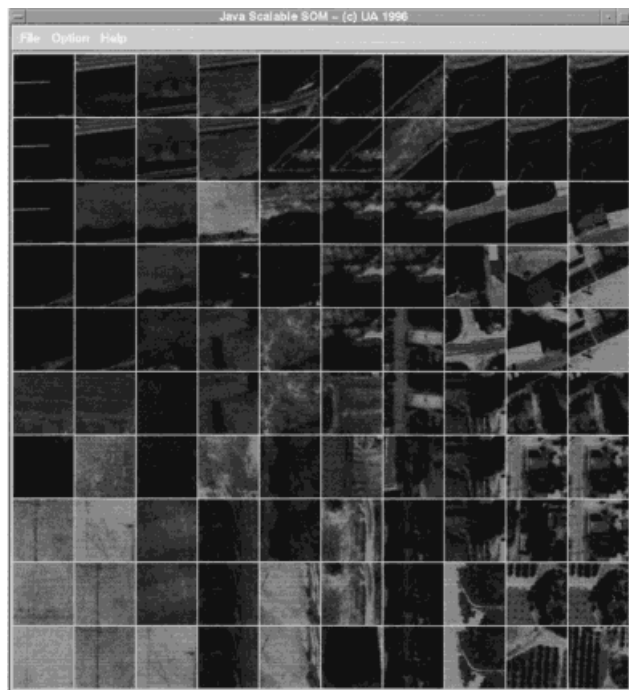


FIG. 3. Region based self-organizing map display.

ability to easily display output nodes as a two-dimensional grid labeled with text or images. Similar nodes are topologically close on the map presenting a visualization that is congruent hopefully with human interpretation as expressed by Small (1999). During training, inputs are presented sequentially with the desired output left unspecified. After training, inputs are mapped to nodes which have connection weights that represent cluster centers in the multidimensional vector space. The connection weights of topologically close output nodes are sensitive to similar inputs and may be combined, based on connection weight similarity, to form regions (Chen, Smith, Larsgaard, Hill, & Ramsey, 1997).

Tile-Based SOM

A node in the neural network consists of a set of 60 floating point numbers—weights, that represent the node's position in the multidimensional feature space and are randomly initiated. The neural network is trained by presenting the feature vector for each tile to all of the nodes and finding the node with the minimum Euclidean distance between its weights and the tile feature vector. The winning node and its neighbors—output nodes that are within a given radius—have their weights adjusted to approach the tile feature vector. This process is done for a given number of iterations with a decreasing learning rate and neighborhood size. Kohonen (1995) and Lippmann (1987) provide a more detailed description of the SOM algorithm.

After the last training iteration, every tile is assigned to a node in the SOM by finding the node with the minimum

Euclidean distance based on its weights and the tile's feature vector. Then, each node is labeled with a tile by finding the tile with the minimum Euclidean distance for each node. Regions in the SOM can be formed by grouping adjacent nodes that are labeled with the same tile. Figure 3 shows a trained SOM that was created by a demonstration program written in Java™. It reads the feature vectors from the file created in the extraction process and clusters them. Subsequently, the tile file is read and the bitmaps are used to display the tiles corresponding to the feature vectors in the SOM. The user can click on any of the tiles that represent the output nodes and the tiles clustered in the node are displayed in a separate window. Selecting a tile in this window will cause the photograph containing the tile to be shown.

Region-Based SOM

The region based SOM uses the same method of creation except region feature vectors are the inputs instead of individual tile feature vectors. Output nodes are also labeled by the closest region instead of the closest tile. Figure 3 shows the output of a trained SOM based on regions that was also written in Java™. The user can select an output node's representative region and see the node's clustered regions in a separate window. Inputs to this process are the region feature file used for clustering and the tile file used for displaying the regions corresponding to the region features.

Texture Thesaurus Browsing

When users invoke the tile based visual thesaurus, they will see a two-dimensional map of the texture classifications

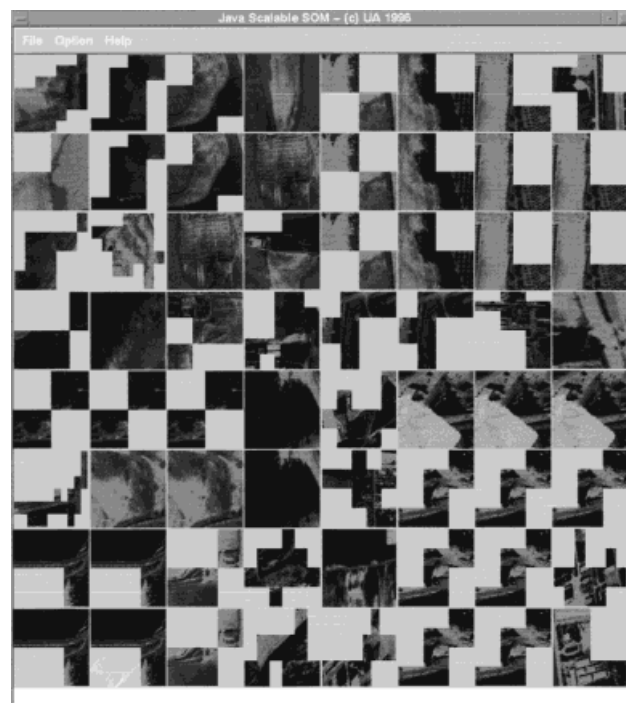


FIG. 4. Tile based self-organizing map display.

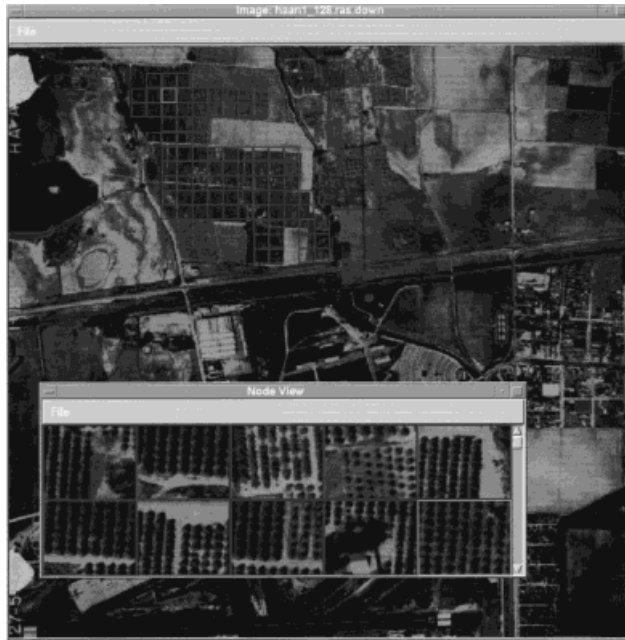


FIG. 5. Image containing tile selected from a cluster in a self-organizing map.

found by the SOM created in the previous section. Figure 4 shows a map with airport runway textures located in the upper-left corner, barren land in the lower-left corner, and vegetation in the lower-right with urban areas located just above it. Other nodes show roadways, intersections, and mixtures of textures.

Users can then select the tile with a texture that is closest to the concept that interests them, such as vegetation. A window similar to the one in the foreground of Figure 5 will appear showing all the tiles that were clustered within that node. Users can again choose a tile that most closely resembles the concept they are seeking. Then, the image containing the selected tile will appear as shown by the background window in Figure 5. The selected tile is highlighted in red in both the tile and image windows. Blue highlighted tiles are those that exist in the retrieved image and are located in the same cluster as the selected tile. In Figure 5 the tiles in a large section of vegetation at the top of the background image were all classified by the currently selected node.

The region based visual thesaurus is similar to the one for textures except a map of regions is displayed from which users select the region that is the most similar to the concept they are searching for. Figure 3 shows a region map with mixed-soil textures in the upper-left corner, shoreline in the lower-left corner, urban area in the bottom-right, and mixed-vegetation/fallow-field areas in the upper-right corner. The foreground window in Figure 6 shows the subsequently displayed window of the regions that were mapped to the SOM output node corresponding to the user selected region of urban areas. After selecting a region in this window, the image containing the region is displayed as shown by the background window in Figure 6. The selected region

is highlighted in red and the other cluster regions in blue. Much of the urban texture tiles at the right side of the image are classified by the node along with some vegetation at the top. The vegetation and urban areas have similar periodic textures resulting in the same classification. Users probably would prefer to have the textures separated which texture based clustering cannot always perform. This problem underscores the need to assimilate other data such as the AVHRR to create more meaningful classifications.

Both the tile- and region-based visual thesauri allow users to find images of interest in very large collections. The images could then be used to feed a query-by-example search engine based on texture to find other similar images. We have not currently created a search utility but this is the next logical step.

Vegetation and Temperature Thesaurus

The goal of generating the vegetation and temperature thesaurus is to allow users to browse surface vegetation and surface temperature information spaces. For this system we use an SOM to classify and display the AVHRR data for Arizona.

Data Set Description

The data used include NASA's Pathfinder AVHRR data and United States Geological Survey's GNIS gazetteer. The GNIS gazetteer consists of about 56,000 place names and their associated coordinates. The AVHRR data is acquired by the National Oceanic and Atmospheric Administration's

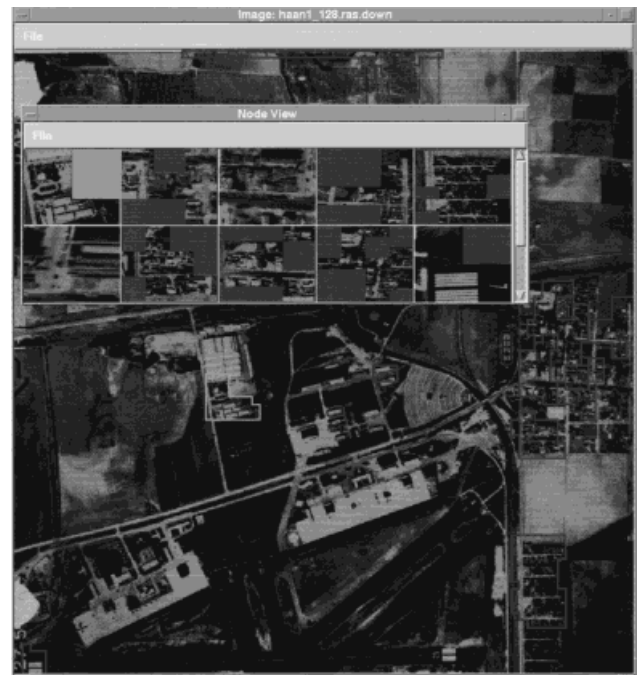


FIG. 6. Image containing region chosen from a cluster in a self-organizing map.

(NOAA) Television Infrared Observation Satellite (TIROS), and is archived and made available for distribution by the United States Geological Survey, Earth Resources Observation Systems, and the European Space Agency. The raw AVHRR data is composed of the afternoon observations over all land and coastal zones from NOAA's polar-orbiting TIROS.

NASA's Pathfinder program determines the latitude and longitude for each pixel of the raw data based on precise navigation information. Atmospheric correction is also applied to remove effects of atmospheric factors, such as water vapor, aerosols, and ozone. The 10-day composite data provided by the Pathfinder AVHRR Land (PAL) data removes the effects of clouds by choosing the observations on the day that has the highest Normalized Difference Vegetation Index (NDVI) value. Therefore the 10-day PAL data can provide the surface observations under near-clear sky condition. The spatial resolution of this data is 8 km.

The AVHRR sensor measures emitted and reflected radiation in five channels (bands) of the electromagnetic spectrum: a visible (0.58 to 0.68 micrometer) band, a near-infrared (0.725 to 1.1 micrometer) band, a mid-infrared (3.55 to 3.93 micrometer) band, and two thermal infrared (10.5 to 11.5 micrometer, 11.5 to 12.5 micrometer) bands. The mid-infrared band is used for sea surface temperature mapping and is not available in the PAL data. The first AVHRR channel is located in a part of the spectrum where chlorophyll in leaves causes considerable absorption of incoming radiation, and the second channel is in a region where the spongy mesophyll leaf structure leads to considerable reflection. The NDVI is calculated by the following formula:

$$\frac{(\text{Channel 2 reflectance} - \text{Channel 1 reflectance})}{(\text{Channel 2 reflectance} + \text{Channel 1 reflectance})}$$

The NDVI equation produces values in the range of 1.0 to -1.0, where increasing positive values indicate increasing green vegetation and the negative values indicate non-vegetated surface features such as water, barren land, ice, and snow. The radiances from channels 4 and 5 depend on the surface temperature. Therefore, by converting the radiance to brightness temperature, an approximation of the surface temperature will be obtained.

According to the goal of this experiment, the PAL data over Arizona in 1993 has been chosen as the test bed. The NDVI data and the information from channels 4 and 5 will be used to form the input of the thesaurus.

AVHRR Thesaurus Creation

Step 1: Feature vector extraction

The Goode Interrupt Homolosine Projection used by the PAL data has to be converted to the equal angular projection so that the latitudes and longitudes of pixels can be determined. Each pixel is assigned a name according to its

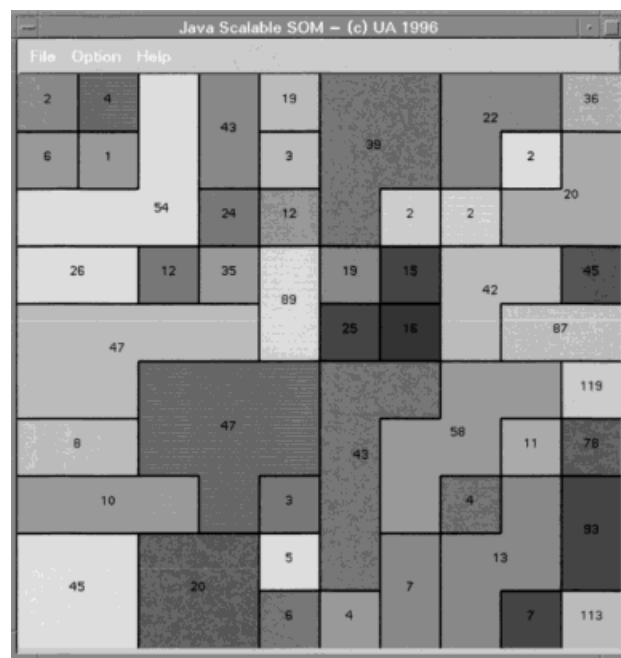


FIG. 7. AVHRR self-organizing map display.

location by using the GNIS gazetteer. Pixels with unusable records or missing values have been discarded and 1442 pixels over Arizona have been chosen. One year (1993) NDVI and thermal temperatures from channel 4 and channel 5 are used to produce feature vectors of 108 floating point numbers. The formed feature vector for each pixel can represent not only the surface vegetation type and temperature but also the seasonal change of the vegetation and temperature.

Step 2: Self-organizing map formation

A 10×10 SOM map has been employed to visualize the categorization of the AVHRR data. The chosen feature vectors of the 1442 pixels were used as the input nodes of the map. The SOM was trained in the same manner as the texture-based maps resulting in the assignment of pixels with similar vegetation type and thermal temperature to the same region.

Figure 7 shows the SOM produced using Java. The numbers on the regions represent number of pixels assigned to the region. When an output region is selected by the user, all the pixels belonging to the region will be displayed on a map of Arizona. Figure 8 is a display of 54 pixel locations from the region at the north-east corner of the map in Figure 7. The pixels near Flagstaff, east of Showlow, and near Alpine, were assigned to this region. According to the SOM theory, all these pixels should have similar surface vegetation type and temperature. This can be rationalized by the fact that the areas of Flagstaff, Showlow, and Alpine are within national forest parks and have the densest vegetation

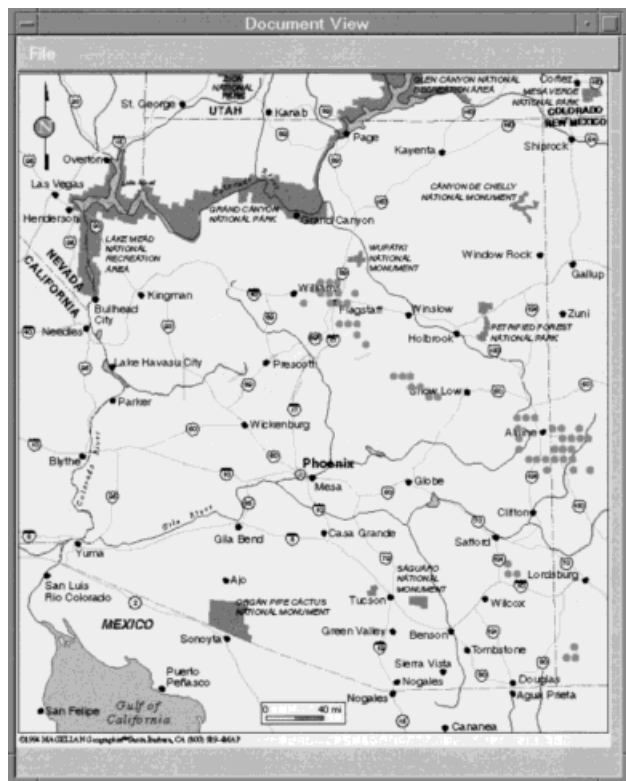


FIG. 8. Map showing pixel locations for a cluster in AVHRR self-organizing map.

cover in Arizona. The areas are also the highest regions in the state thus having the lowest temperatures.

Visual Thesaurus Interface for the AVHRR Data

The vegetation-temperature patterns have been clustered on a graphical two-dimensional display (SOM map). Pixels with similar vegetation and surface temperature are grouped in the same region of the display. For instance, the green/cold patterns are at the upper-left corner of Figure 7, while the desert/warm patterns are at the bottom-right corner. When a vegetation/temperature pattern on the display is clicked, pixels belonging to this pattern will be displayed on an Arizona map in another window.

This prototype system will enable users to find the number of vegetation patterns in Arizona. Also, users will have little problem differentiating patterns because each pattern has its own color. The four corners of the SOM in Figure 7 are occupied by four typical vegetation/temperature types (i.e., green/cold pattern at top-left, green/warm pattern at bottom-left, desert/warm pattern at bottom-right, and desert/cold-pattern at top-right). The other areas on the display are occupied by mixed type patterns. The distribution of the patterns in the map will help users to efficiently locate interesting patterns.

Instead of displaying the place names when a pattern is clicked, the system displays the locations of the pixels on a map of Arizona shown in Figure 8. The display shows the

geological positions of the pixels which are more meaningful to users who are unfamiliar with the place names. Additionally, the interface presents the distribution of the selected pattern over the state of Arizona. If the users are familiar with certain places, they can find other places with the similar vegetation/temperature pattern by looking at the pixels on the map. Therefore, users will still be able to find interesting places even though they are not familiar with the area's pattern.

This system will answer queries like "find all the places with a lot of green and low temperature in Arizona" or "find places which have similar temperature and vegetation as Tucson." Users will be able to browse places according to the vegetation/temperature pattern. Some other information, like aerial photographs, vegetation type data, and satellite images may be used jointly with AVHRR data to answer various geological and environmental queries.

AVHRR Discussion

The SOM map appears to be a powerful method for thesaurus formation and visualization. The combination of AVHRR data and the SOM map will be able to answer queries concerned with the surface vegetation and surface temperature. Future work will include conducting an experiment to evaluate the performance and usefulness of the thesaurus; combining AVHRR data with other data sets, such as Digital Elevation Models data; and answering other climate or land cover queries.

Implementation Details

The feature extraction, tile file creation, and region segmentation programs are all written in C by B. S. Manjunath's group at the University of California, Santa Barbara. We have made some modifications to the feature extraction program to parallelize it for execution on SGI's 64 node Origin 2000 supercomputer at NCSA. Feature extraction of all the tiles in an image, typically 64,000 tiles, is the most computational intensive part of the process. By parallelizing the extraction, we are able to reduce the time linearly with respect to the number of nodes used on the Origin 2000.

The rest of the programs, including the SOM and image display, were written in Java™, an object-oriented language created by Sun Microsystems and modeled after C++. Java™ is "small, simple, and portable across platforms and operating systems, both at the source and binary level" (Lemay & Perkins, 1996). It is compiled and interpreted, meaning the Java™ compiler converts source code to platform independent byte code (Flanagan, 1996). This byte code represents instructions for the Java™ virtual machine which is easily emulated on most modern hardware platforms. Ports currently exist for Windows NT, Windows 95, Macintosh, HP-UX, Solaris, Digital UNIX, Linux, and others. The portability does come at a cost in performance since the byte code is interpreted using the virtual machine. However, the performance difference has been significantly re-

duced because many of the available Java™ ports include a just-in-time (JIT) compiler that converts the Java™ byte code to native code while the program runs (Flanagan, 1996).

We chose to implement the user interface for the visual thesaurus in Java™ because of the language's portability and rapid graphical user interface (GUI) development features. We were not greatly concerned with performance because it is not usually an issue for GUIs which spend most of their time waiting for user input.

The demonstration programs are not just GUIs but also rely on a clustering component implemented as an SOM for which performance is a concern. In our early prototypes, the SOM's performance was more than adequate mainly due to the implementation of our scalable SOM algorithm (Rousinov & Chen, 1998). The JIT compilers, while not included in Sunsoft's standard Java™ Development Kit but available on SGI workstations and the Origin 2000, increase the performance to near compiled C code's speed. Additionally, we have created a C++ implementation of the SOM that we are currently testing and will parallelize for greater execution speed.

Since we foresee the visual thesaurus as executing in a distributed computing environment, security is an issue. Programs received over a network must be able to guarantee safe execution without virus contamination or Trojan horses. An attractive feature of Java™ is its ability to address many security issues as discussed by Niemeyer and Peck (1996).

A useful interface must be easily accessible and an effective means to provide accessibility is the leveraging of Internet technology. The latest versions of many Web browsers incorporate the Java™ virtual machine enabling them to load programs from remote locations and run them locally as embedded Java™ applets. The incorporation of Java™ in Web browsers allows a large number of people to use programs written in the language. Thus, Java™ applications have a larger potential user base than ones created by compiled languages such as C or C++ that are platform dependent.

Future visual thesauri could also be implemented using client-server architecture. Clients would connect to the server which would provide images used in the search. Java™ is an excellent choice for both the client and servers because it has significant advantages over traditional Common Gateway Interface (CGI) programs.

Homepages with dynamic content have traditionally been produced by CGI programs that provide a means of connecting client programs with information servers using the Web HyperText Transport Protocol (HTTP). When users access pages that use CGI programs, their browsers submit requests to the HTTP servers which respond by performing actions, such as creating new homepages. However, the communication between the users' browsers and the HTTP servers is not dynamic. The servers are completely event driven, meaning they can only respond to requests from the clients and cannot initiate requests of their

own. In addition, CGI programs can only respond to one event before terminating. These limitations lead to relatively static user interfaces for CGI-based applications. This is especially frustrating when frequently used requests take more than a few moments to satisfy, causing users to wonder if the program is still running (Beard, 1991). Java™'s ability to avoid these pitfalls fortified our decision to use the language.

Conclusion

The scalable algorithms we have used to automatically create thesauri of texts can be applied to images when combined with textural analysis methods. Kohonen's SOM and Gabor filters appear to be a powerful combination for visual thesauri generation of aerial photographs. The thesaurus of AVHRR data also appears promising and the data may be used to augment the texture feature vectors to build an enhanced SOM of geo-spatial media. However, before any conclusions can be made about the visual thesauri's utility, a usability study needs to be conducted.

The use of Java™ to build the interfaces is a solid technique for creating dynamic web accessible applications since the language is powerful, portable, and secure. The only problem we encountered with Java™ is that the development kits for different platforms are not yet consistent.

Future Work

In the future we plan to use the texture clusters found by the SOM as classifiers to index images and perform co-occurrence analysis. Each texture region of an image is mapped to an output node in the SOM representing a unique cluster of similar textures. The output node number can be treated as an index for the image texture region and used to compute co-occurrence with other texture types (i.e., textures mapped to other output nodes). Then users can browse the visual thesaurus and find a texture to start their query. A list of other classes of textures frequently co-occurring with the specified one will be displayed in the order of decreasing frequency. From this list the users can refine the queries by selecting other textures they feel are relevant to their search. This process is iterative. Finally the user can browse the images that contain all or most of the selected texture types. The advanced query visualization technique presented by Brooks & Campbell (1999) could be well suited for a texture retrieval system.

Query by example is another exciting possibility that could be realized by creating an SOM based on the image indexes. A sample photograph provided by the user, perhaps found through browsing the visual thesaurus, could be broken down into a set of texture types. Then the image is presented to the SOM to find the cluster of similar images. For example, if the user found a photograph that contained orchards near a river and industrial park, they could use it to find other images containing similar combinations of orchards, rivers, and industrial parks.

These methods build on our experience with co-occurrence analysis, and query refinement, and category maps of texts. We believe our past success should carry over to geo-spatial data types and plan to pursue this path.

Acknowledgment

This project is supported mainly by the following grants:

- NSF/ARPA/NASA Digital Library Initiative, 1996–1998 (H. Chen and T. Smith, “Supplement to Alexandria DLI Project: A Semantic Interoperability Experiment for Spatially-Oriented Multimedia”)
- NSF/ARPA/NASA Digital Library Initiative, IRI-9411318, 1994–1998 (B. Schatz, H. Chen, et. al., “Building the Interspace: Digital Library Infrastructure for a University Engineering Community”)
- NSF CISE, IRI-9525790, 1995–1998 (H. Chen, “Concept-based Categorization and Search on Internet: A Machine Learning, Parallel Computing Approach”)
- National Center for Supercomputing Applications (NCSA), High-performance Computing Resources Grants, 1994–1996 (H. Chen).

The authors thank the Distributed Active Archive Center (Code 902.2) at the Goddard Space Flight Center, Greenbelt, MD, 20771, for producing the data in their present form and distributing them. The original data products were produced under the NOAA/NASA Pathfinder program, by a processing team headed by Ms. Mary James of the Goddard Global Change Data Center and the science algorithms were established by the AVHRR Land Science Working Group, chaired by Dr. John Townshend of the University of Maryland. Goddard’s contributions to these activities were sponsored by NASA’s Mission to Planet Earth program.

References

- Beard, D.V. (1991). Computer human interaction for image information systems. *Journal of the American Society for Information Science*, 42(8), 600–608.
- Brooks, M. & Campbell, J. (1999). Interactive graphical queries for bibliographic search. *Journal of the American Society for Information Science*, 50, 814–825.
- Carlson, B. (1997). Web still and video image content search and retrieval. *Advanced Imaging*, 60, 34–36.
- Chen, H., Schuffels, C., & Orwig, R. (1996). Internet categorization and search: A machine learning approach. *Journal of Visual Communications and Image Representation*, 7(1), 88–102.
- Chen, H., Smith, T.R., Larsgaard, M.L., Hill, L.L., & Ramsey, M. (1997). A geographic knowledge representation system (gkrs) for multimedia geospatial retrieval and analysis. *International Journal of Digital Libraries*.
- Flanagan, D. (1996). *Java in a nutshell* (Chap. 1). Sebastol, CA: O’Reilly & Associates.
- Gabor, D. (1946). Theory of communication. *Journal of the Institution of Electrical Engineers*, 93, 429–459.
- Kohonen, T. (1995). *Self-Organized Maps* (Chap. 3). Berlin, Heidelberg: Springer-Verlag.
- Lemay, L. & Perkins, C.L. (1996). *Teach Yourself Java in 21 Days* (Chap. 1). Indianapolis, IN: Sams.net.
- Lin, X. (1997). Map displays for information retrieval. *Journal of the American Society for Information Science*, 48(1), 40–54.
- Lippmann, R.P. (1987). An introduction to computing with neural networks. *IEEE Acoustics Speech, and Signal Processing Magazine*, 4(2), 4–22.
- Ma, W.Y. & Manjunath, B.S. (1996). A pattern thesaurus for browsing large aerial photographs. ECE Technical Report #96-10, Santa Barbara, CA: University of California Santa Barbara. Available at: http://vivaldi.ece.ucsb.edu/papers/abstracts/bsm_ma_icip96.htm.
- Manjunath, B.S. & Ma, W.Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8), 837–841.
- Niemeyer, P. & Peck, J. (1996). *Exploring Java* (Chap. 1). Sebastol, CA: O’Reilly & Associates.
- Picard, R.W. & Kabir, T. (1993). Finding similar patterns in large image databases. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing* (Vol. V, pages 161–164), New York: IEEE.
- Rorvig, M.E. (1990). Introduction. *Library Trends*, 38(4), 639–643.
- Rorvig, M.E., Turner, C.H., & Moncada, J. (1999). The NASA image collection visual thesaurus. *Journal of the American Society for Information Science*, 50, 794–798.
- Roussinov, D. & Chen, H. (1999). A scalable self-organizing map algorithm for textual classification: A neural network approach to thesaurus generation. *Communication Cognition and Artificial Intelligence*, 15(1–2), 81–112.
- Small, H. (1999). Visualizing science by citation mapping. *Journal of the American Society for Information Science*, 50, 799–813.