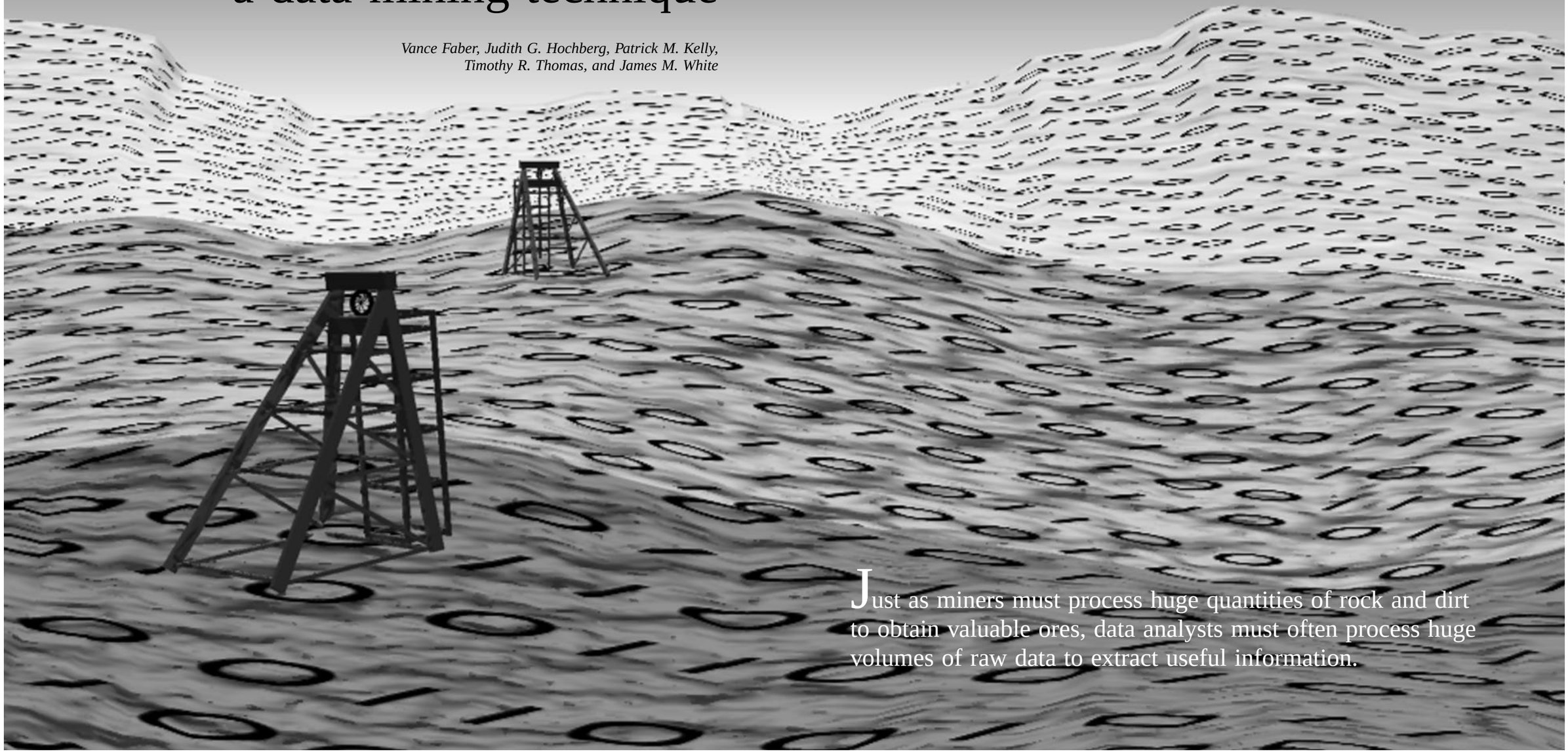
CONCEPT EXTRACTION
a data-mining technique

Vance Faber, Judith G. Hochberg, Patrick M. Kelly,
Timothy R. Thomas, and James M. White



Just as miners must process huge quantities of rock and dirt to obtain valuable ores, data analysts must often process huge volumes of raw data to extract useful information.

The use of computers in numerous applications is generating data at a rate that far outstrips our ability to process and analyze it. For example, NASA satellites are expected to generate hundreds of terabytes of data per day (1 terabyte = 10^{12} bytes). Sets of financial data ranging from credit-card transactions to shipping records contain terabytes, and textual databases are growing rapidly. A great deal of effort is currently being expended to develop new hardware and software to generate, transmit, and store such data, but relatively little emphasis has been placed on developing new ways to use computers to analyze the data after they are acquired.

"Data mining," or the process of extracting useful information from very large datasets, is a focus of our efforts in the Laboratory's Computer Research and Applications Group. In this article we describe a data-mining method we call concept extraction, which involves both humans and computers in a productive partnership. Here "concept" is defined as a psycholinguistic category that can be either physical (*animal* or *blue*) or abstract *(mood* or *politics*).

The underlying premise of concept extraction is that in order to interpret data, humans naturally and quickly extract and identify significant concepts. A person can contemplate a landscape, receive data through the sense organs, and can then make a reasonable judgement about whether it will rain, for example, or whether it will snow. A person can scan a magazine's table of contents and easily select the articles that relate to a subject of interest. Humans are constantly assimilating, categorizing, synthesizing, and analyzing data—most often without any conscious realization of doing so.

The ability to extract concepts from sensory data is the product of millions of years of evolution and years of individual learning and experience. But humans go beyond simply recognizing and naming objects or events; we make judgements based on an overall context or quite subtle correlations among diverse elements of the available data. The great complexity, subjectivity, and ambiguity of human concepts make them extremely difficult if not impossible to define in a quantitative manner appropriate for use by computers. As linguist William Labov puts it, "Words that are bound to simple conjunctive definitions will have little value for application in a world which presents us with an unlimited range of new and variable objects for description."

Computers can, however, make data mining easier because they can quickly and accurately perform certain tasks that demand of humans too much time or concentration. For example, an expert in the interpretation of satellite images may wish to determine the ratio of urban to rural land use in a particular region. No matter how skilled the expert may be at distinguishing between the urban and rural areas in the image, the task of identifying each area in a large image is time-consuming and tedious. In contrast, computers—once appropriately programmed—are ideally suited to that task. We have accelerated the concept-extraction process by breaking it down into quantitative and intuitive portions, and assigning the former to computers and the latter to human experts.

First, the computer reduces the size of the dataset composing the satellite image while retaining its essential character. This crucial step employs a new, high-speed clustering algorithm specifically developed to handle very large datasets. The human then identifies, or extracts, an example of each concept of interest, in this case an urban area and a rural area, within the image produced from the reduced dataset. An expert ability such as this cannot readily be translated into a computer program, but after examples have been provided, the computer can identify any repetitions of the examples in the image.

Our method greatly increases the rate at which an expert, or even a novice, can analyze a large and complex dataset. Concept extraction is not an exact process—even an expert can be inconsistent or make mistakes. But by enabling the computer to approximate the expert's interpretive skills, concept extraction provides a flexible, rapid way to incorporate the human perspective—flaws and all—into computer analysis.

Concept Extraction Applied to Satellite Images

Traditional analysis methods. Images produced by satellites orbiting Earth serve a variety of purposes including assessing weather conditions, such as heavy rains likely to cause flooding, and tracking long-term environmental trends, such as deforestation. Satellite images are, of course, also a source of military intelligence. Here we focus on the application of concept extraction to data obtained by the Landsat 4 system, an unclassified system that has been in operation since 1982.

Traditional image analysis requires an expert for two reasons. First, the analyst must interpret Earth data from a rather unusual perspective—imagine the difference between an eye-level view of a forest and a satellite view from an altitude of 450 miles. Second, the analyst must understand the significance of all the recorded information, which consists of the intensities of the radiation reflected from or emitted by the surface of the earth. Intensity measurements are recorded for seven regions of the electromagnetic spectrum, three

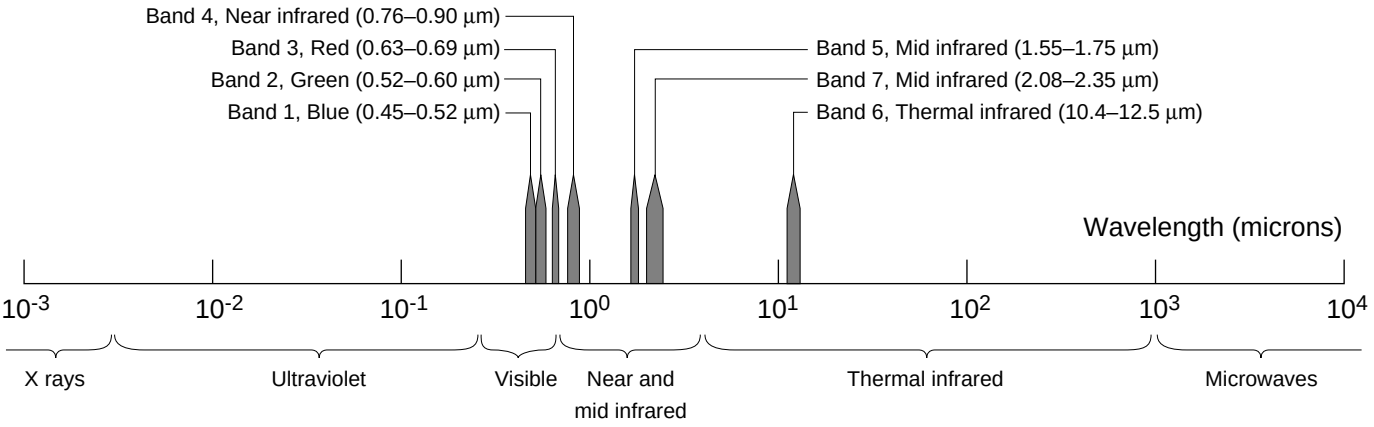


Figure 1. Locations within the Electromagnetic Spectrum of the Seven Bands Recorded by the Landsat System
A portion of the electromagnetic spectrum, the continuum of all electromagnetic waves, is arranged from left to right according to increasing wavelengths. Landsat sensors collect data in seven spectral bands: three in the visible portion of the spectrum (Bands 1, 2, and 3); one in the near-infrared (Band 4); two in the mid-infrared (Bands 5 and 7); and one in the thermal infrared (Band 6).

in the visible region (red, blue, and green) and four in the infrared region (see Figure 1). Analysts must know which spectral regions help identify which surface features and must locate and identify each separate occurrence of each concept. Even an expert analyst may need up to several weeks to extract the desired information from a single Landsat 4 image.

The slow pace of the analysis is mainly due to the enormous size of the dataset. A typical Landsat image covers an area of about 10,000 square miles, or 100 miles on a side. The digital image is composed of about 50 million pixels, or 7000 pixels per side, so each pixel represents a square region of about 75 feet on a side. For each pixel the Landsat system measures the radiation intensity in each of the seven spectral bands and stores each of the measured intensities as an 8-bit (or 1-byte) number. So, the data for a single image amount to 7 1-byte numbers per pixel, or a total of roughly 350 million bytes of data.

These data are typically viewed on a high-cost, 24-bit-per-pixel color screen. To produce an ordinary color image, the red, green, and blue 8-bit intensity values for each pixel of the recorded image are mapped to the blue, green, and red components that constitute each pixel on the screen (see Figure 2). To gain a different perspective and thus additional information, analysts often display other combinations of spectral bands (see Figure 3).

Even a simple mapping of the spectral data to the display screen requires time-consuming computation. The color value for each pixel must be computed separately and recomputed each time a new set of three bands is chosen for display. Usually only a quarter of an image is processed at a time, but processing even 12 million pixels takes long enough—up to a few minutes—that interactive use of the data is impractical. For more complex mappings, where two or more spectral bands might be combined according to some useful function, computations can take up to a few hours.

Data clustering—the first step in concept extraction. One of our goals in developing the concept-extraction technique was to facilitate interactive analysis of Landsat data by reducing the time required for analysis from days to hours.

The essential first step in achieving that goal was to reduce the size of the original dataset by applying a newly developed, high-speed computer algorithm for clustering the data. (See "Clustering and the Continuous *k*-Means Algorithm.") Clustering allows us to replace the original spectral data with an appropriate set of representative values. To find those values, the seven intensity values for each of the 50 million pixels are regarded as the components of a vector in a seven-dimensional spectral-intensity space. That is, each axis in the space corresponds to one of the seven spectral bands, and the coordinates specifying the end point of each vector are the seven spectral intensity values of the pixel that vector represents. The *k*-means algorithm groups the 50 million points into *k* clusters such that all the points in each cluster are more similar ("closer") to one another than to those in the other clusters. For the Landsat data we chose a *k* value of 256 because 256 is the greatest number of distinct colors that can be represented on an inexpensive, 8-bit screen; we often use larger *k* values for other applications. The algorithm also determines each cluster's centroid, which is the cluster's "mean point," or the point each of whose coordinates is the arithmetic average of the corre-

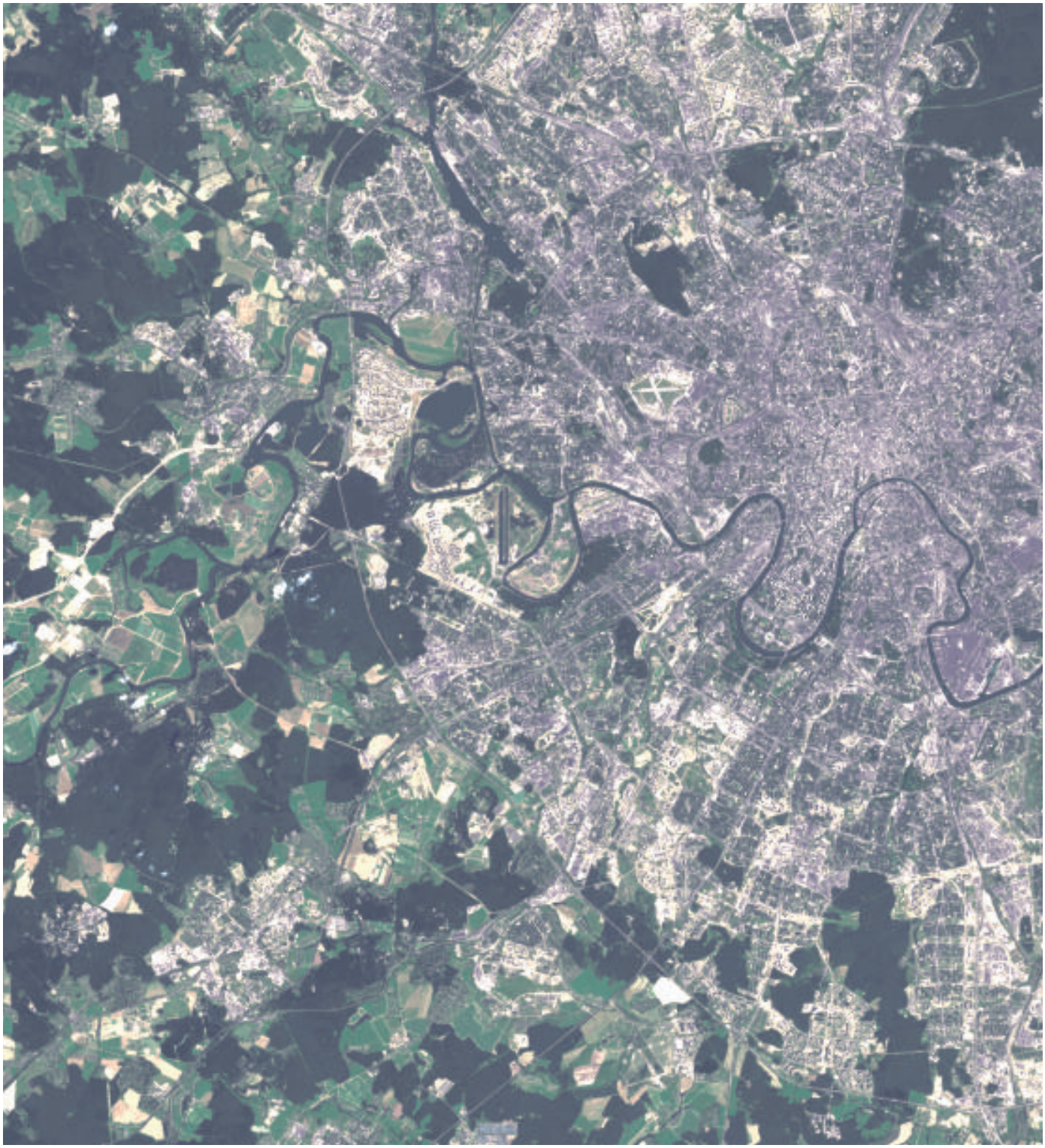


Figure 2. Landsat Image of Moscow and Its Environs

This image of Moscow and environs is based on data obtained by a Landsat satellite late in the growing season. The satellite, in orbit at an altitude of 450 miles, measured the intensities, in seven spectral bands, of the radiation reflected from or emitted by the surface of the earth. This is a small portion of the entire 50-million-pixel image, only about 800,000 pixels, and each pixel represents a square region of about 75 feet on a side. The image is an ordinary, or "natural-color," image, and the color of each pixel is determined by combinations of intensity values in the blue, green, and red spectral bands mapped to the blue, green, and red components of each pixel. However, the intensity values combined to produce the image shown are not those measured by the satellite but, rather, are those resulting from a data-reduction technique called clustering, as discussed in the main text. The image is reproduced as it would appear on an 8-bit-per-pixel color screen.

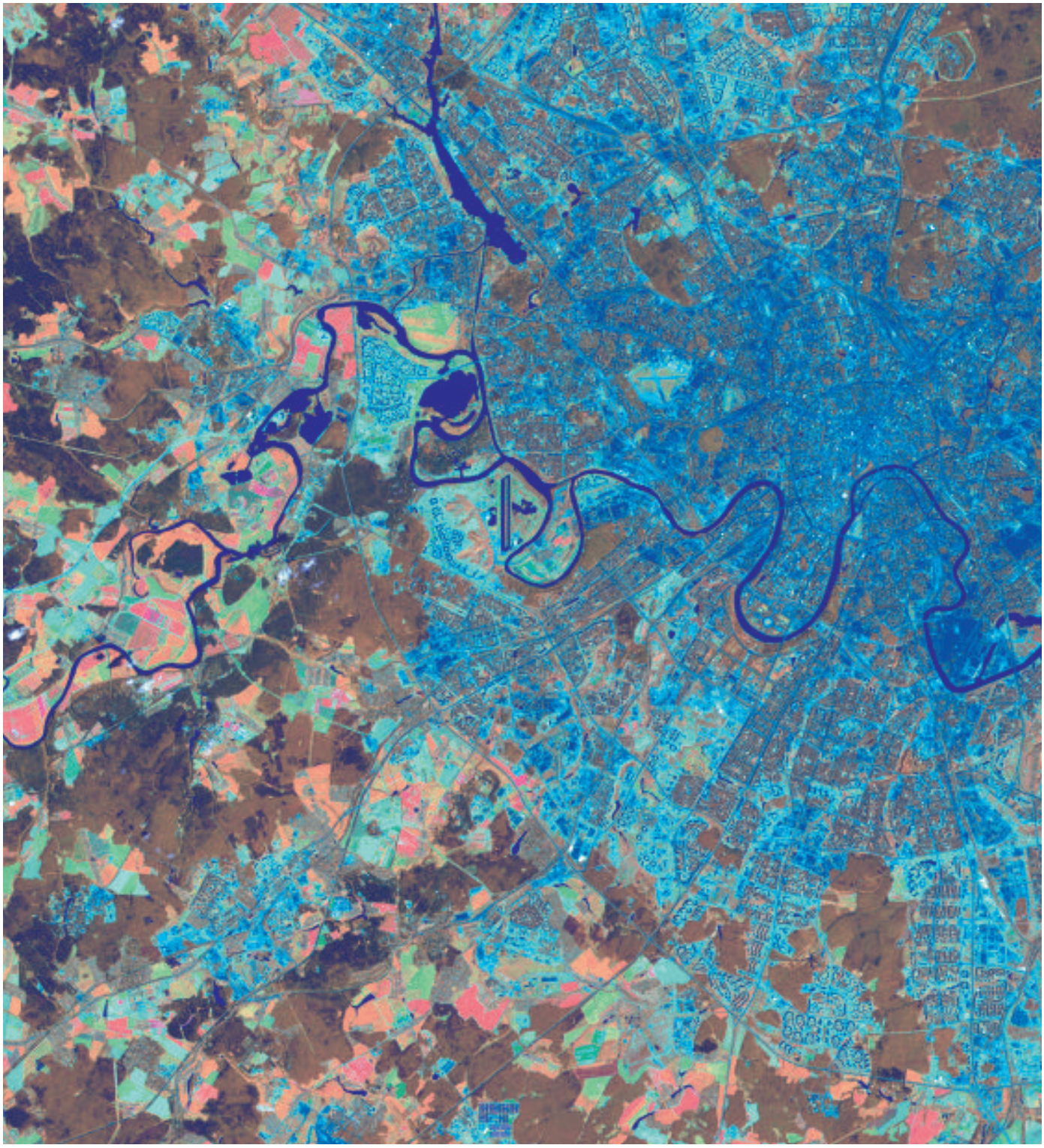


Figure 3. An Alternative Imaging of Moscow and Its Environs

This image was produced from the same reduced dataset that yielded the image in Figure 2. Here, however, is an alternative representation that yields optimal visual separation of the various types of vegetation and, in particular, agricultural versus non-agricultural regions. The color of each pixel was determined by a combination of intensities from the near-infrared Band 4, mid-infrared Band 5, and the visible Band 3 (red) mapped to the red, green, and blue pixel components respectively. With this band combination, vegetation types are differentiated by variations in both color and color intensity. The regions of various shades of red and orange are fields of growing crops. Regions with the healthiest vegetation, or greatest amount of "biomass," appear in the most intense shades of red. The pale blue regions are unplanted fields. The brown regions represent forested areas; deciduous forests are light brown, and coniferous forests are a darker shade of brown.

sponding coordinates of all the points in that cluster. Figure 4 illustrates the process of clustering spectral-intensity data for a simplified case that considers only two bands, red and blue, and thus a spectral-intensity space of only two dimensions.

After the Landsat data is grouped by the algorithm into 256 clusters, each cluster is designated by a 1-byte number from 0 to 255 and those designations are stored in a codebook along with the seven intensity values for the centroid of each cluster. Each pixel in the image is now associated with, or belongs to, a spectral cluster—the cluster into which its spectral data have been grouped. Furthermore, each pixel will now be "colored" according to the intensity values of that cluster's centroid rather than according to the original spectral data. Figure 5 illustrates the stored data array before and after clustering.

The centroid data stored in the codebook are used to approximate the original spectral data. To produce a color image, any combination of up to three of the seven spectral bands—or any mathematical transformation of all seven—are mapped to the three components of each pixel on a color screen. Although clustering drastically reduces the amount of spectral data to the 256 centroid values, that number is substantially greater than the number of concepts (urban, rural, forest, desert, and so on) we intend to extract from the data. Thus the data still have enough detail to allow clear distinctions among those concepts. In fact, the human eye cannot distinguish a display of the clustered data from a display of the original data.

Storage and handling of clustered data. The use of clustered data greatly enhances the efficiency of all future data handling. Since the amount of

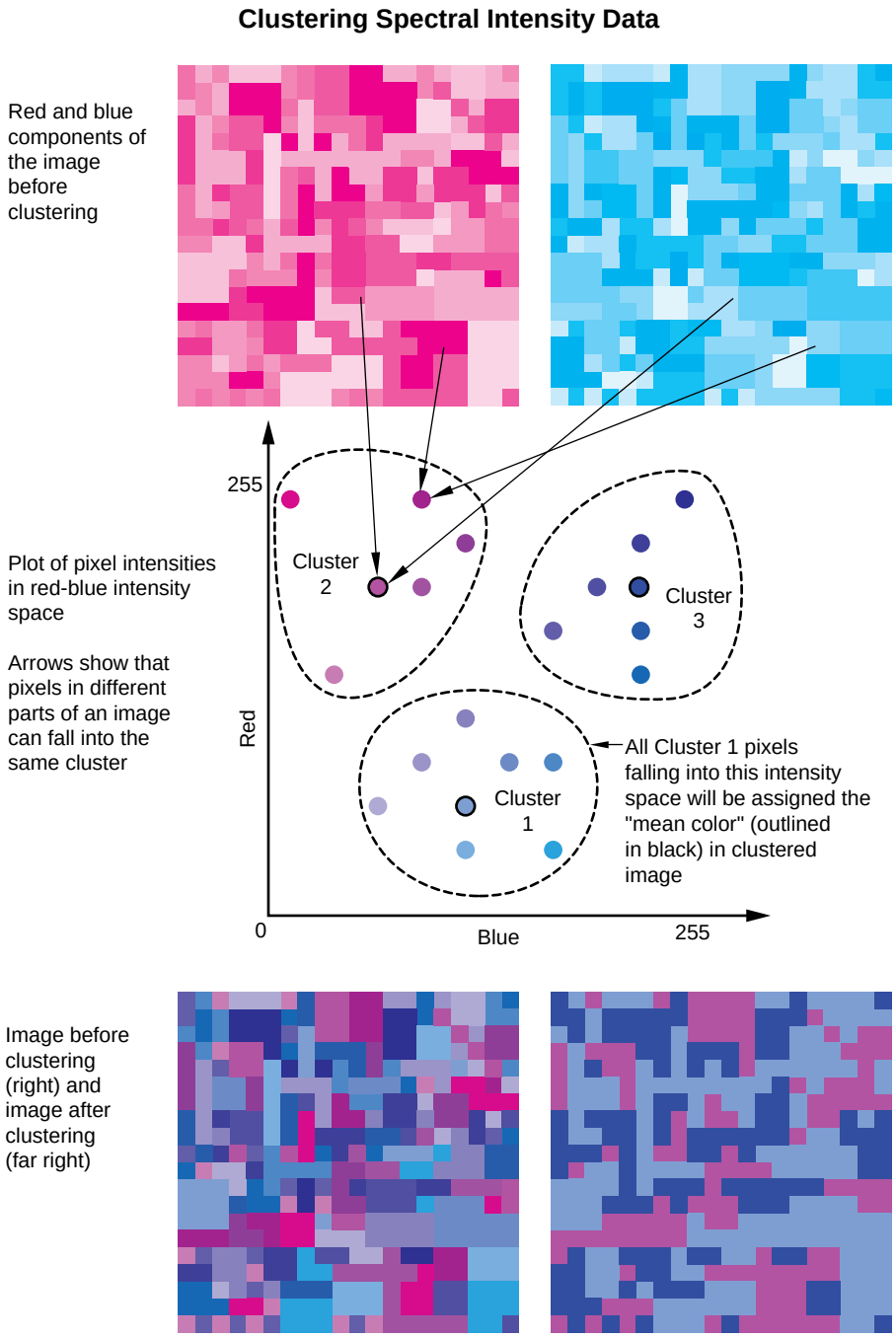


Figure 4. Clustering Landsat Data

The figure illustrates the clustering of spectral-intensity data for a simple two-band image. The red and blue components of the digital image are shown separately. Each pixel is plotted in "spectral-intensity space"; that is, the coordinates of each pixel in that space are the red and blue intensities of that pixel in the original image. The data in spectral-intensity space have been grouped into three clusters. Each of the three points outlined in black represents the centroid of its cluster, that is, the average of the coordinates of the data points in each cluster. Thus the color given by the centroid's coordinates is the "mean color" of the cluster. When the image is displayed after the spectral data have been clustered, each pixel that belongs to cluster 1 in spectral-intensity space is colored with the centroid color of cluster 1, and similarly the pixels that belong to other clusters are colored with the centroid color of their clusters. Thus in this simplified example the final image is composed of pixels with just three colors. In reality, the data for Landsat images are grouped into 256 clusters, and the resulting 256-color images cannot be visually distinguished from the original Landsat images.

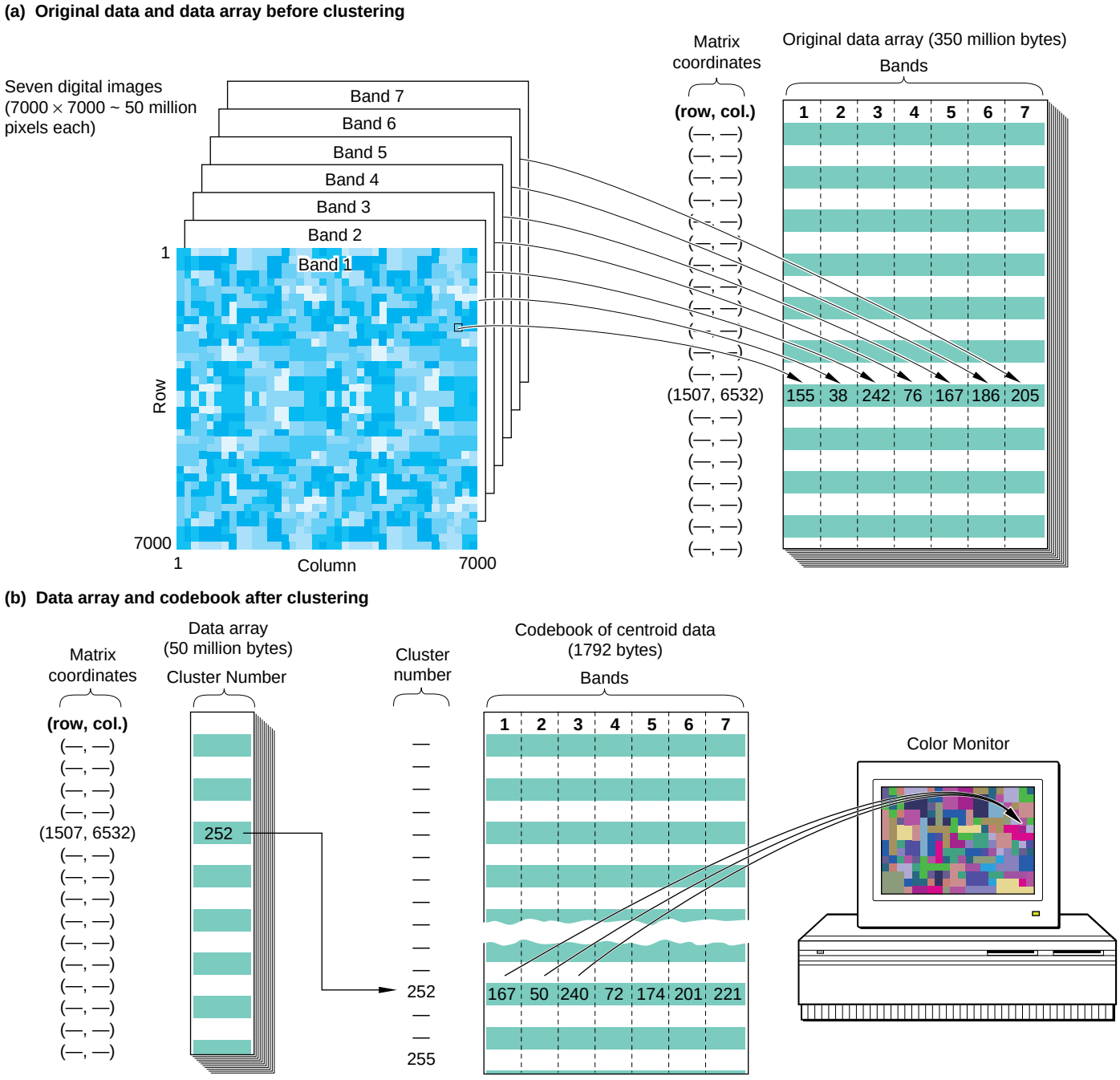


Figure 5. The Data Array before and after Clustering

(a) The original data for a Landsat image are represented as a set of seven digital images, one for each of the seven spectral bands recorded by the Landsat system. The location of each of the roughly 50 million pixels composing each image is specified by its matrix coordinates. Also shown is the stored data array for the entire data set prior to clustering. The array contains the seven 1-byte numbers that correspond to the seven different spectral intensities recorded at each pixel location. Thus, prior to clustering, the spectral data for a single Landsat image occupy roughly 350 million bytes of memory. The two components of the stored data array after clustering are shown in (b). The first component is an array consisting of the pixel cluster numbers. (A pixel's cluster number is a 1-byte number that specifies the cluster into which the spectral data for that pixel have been grouped.) The second component is the codebook, or lookup table, which contains the seven spectral values of the centroid of each of the 256 clusters. After clustering, the mean spectral data in the codebook replace all the original spectral data, and the clustered data occupy only about 50 million bytes of memory. As shown in the figure, the cluster number of each pixel links the pixel's coordinates to the appropriate centroid data in the codebook. Those data specify the spectral characteristics of that pixel after clustering, and the figure shows how the clustered data are mapped to an 8-bit-per-pixel color screen. Any combination or mathematical transformation of the seven spectral bands can be used to create the color image. The computer accesses the codebook to find the appropriate spectral values for each pixel.

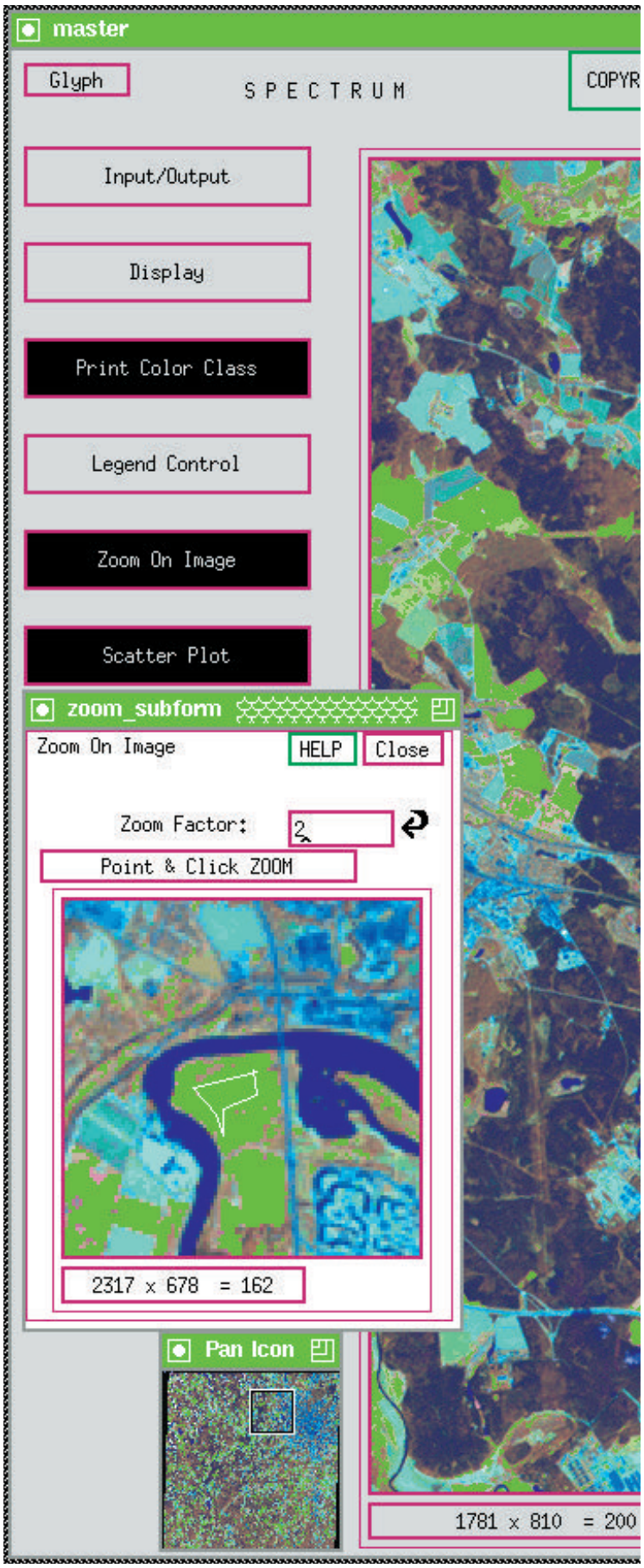
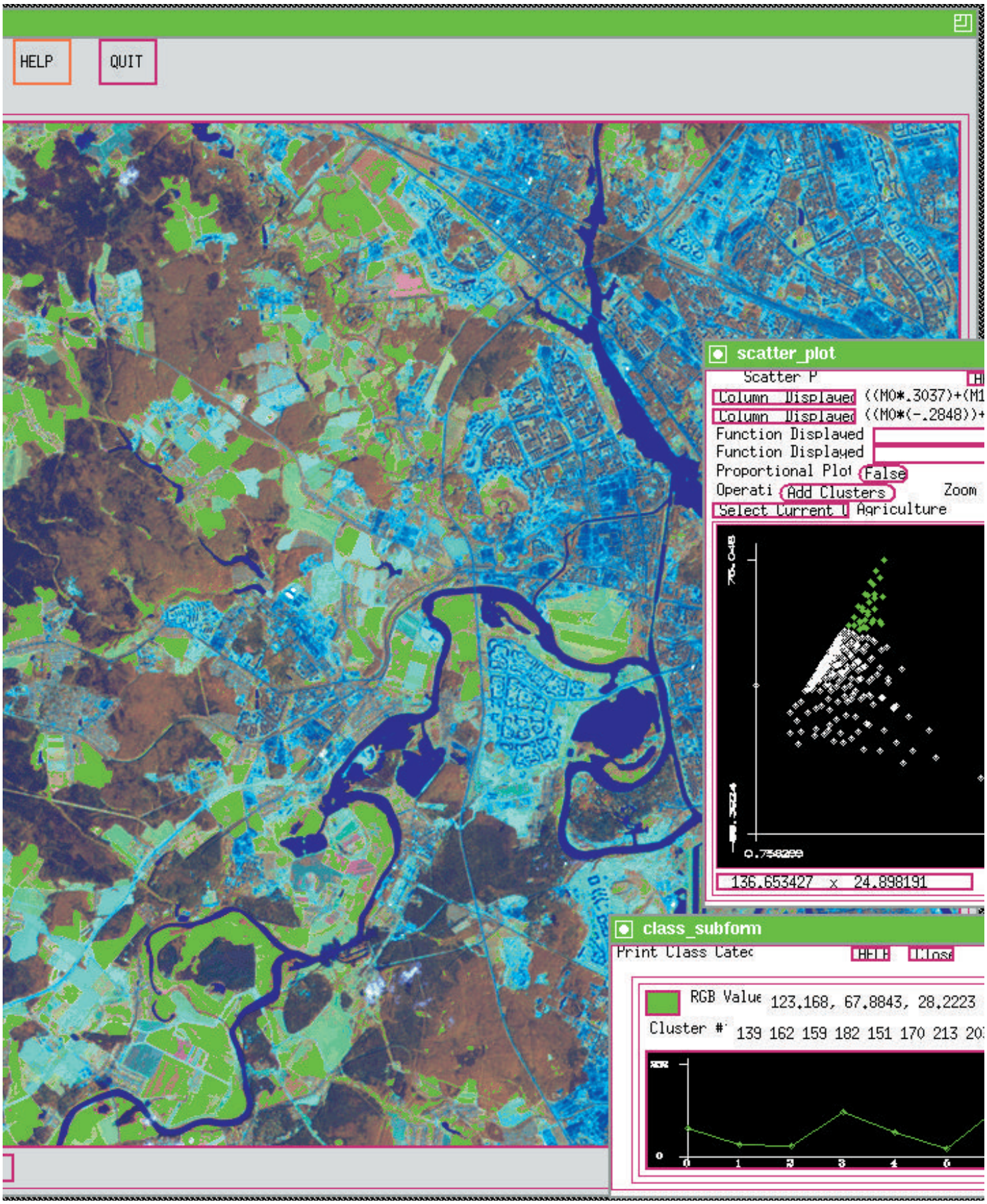


Figure 6. Using Spectrum to Extract Concepts

The figure shows the window environment provided by Spectrum, a software package developed through a collaboration between the Laboratory and the University of New Mexico. The small window in the lower left-hand corner of the screen shows the entire quarter-scene image. The large center window shows a smaller region at full resolution. This region was selected by placing a box icon within the small window to enclose a particular region. The medium-sized window on the left is a magnification of a region selected by placing a cursor within the center image. Here, the concept-extraction technique has already been applied (as described in the main text). The user drew a polygon within a region identified as an agricultural field, labeled the concept "agriculture," and selected a shade of bright green as the color for that concept. As a result, all of the agricultural fields in the image (the red-orange regions in Figure 3) have been automatically identified and colored a shade of bright green. The scatter_plot window (upper right) and the class_subform window (lower right) are discussed in Figures 7 and 8.


HELP
QUIT
scatter_plot
Scatter P
Column Displayed ((M0*.3037)+(M1
Column Displayed ((M0*(-.2848))+
Function Displayed
Function Displayed
Proportional Plot False
Operati Add Clusters
Zoom
Select Current C Agriculture
136.653427 x 24.898191
class_subform
Print Class Catec
HELP
Close
RGB Value 123.168, 67.8843, 28.2223
Cluster # 139 162 159 182 151 170 213

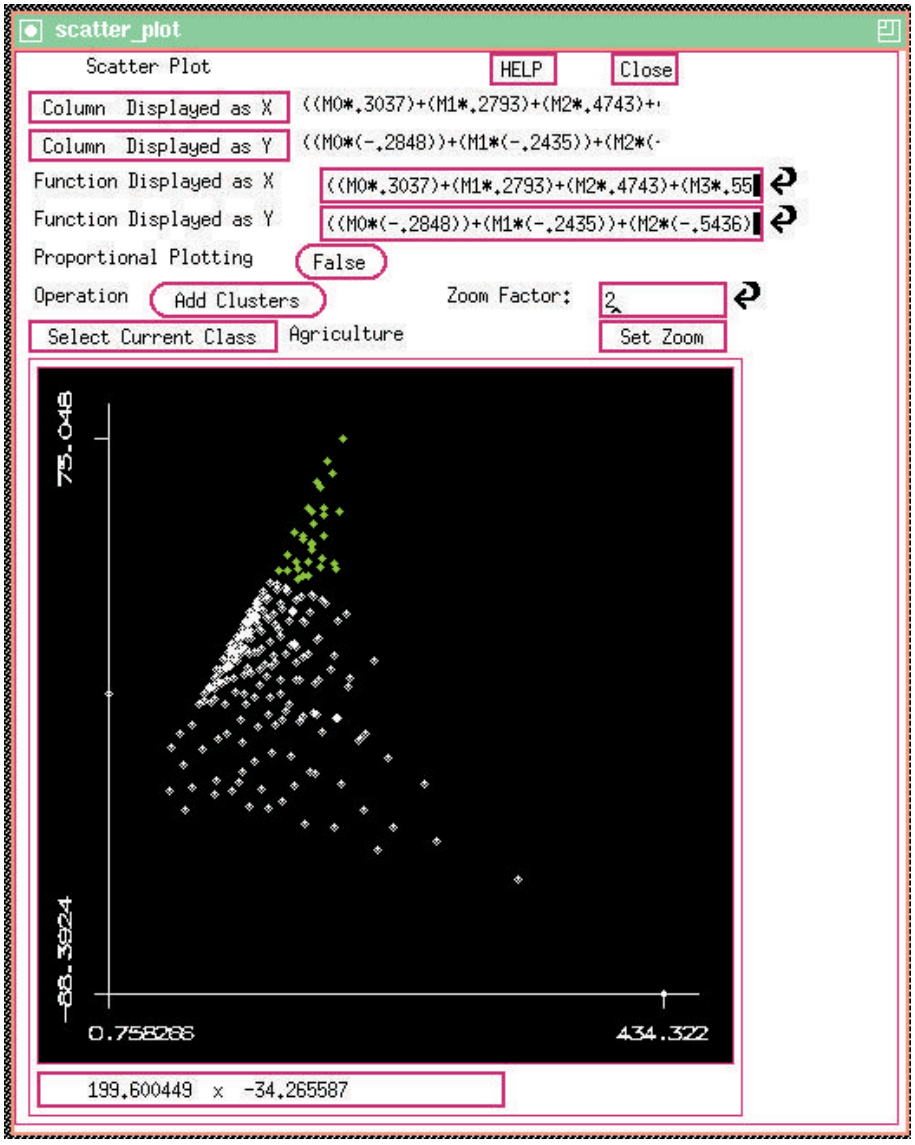


Figure 7. The Scatter_plot Window

The scatter_plot window allows the user to see how clusters within a concept relate to one another as well as to other clusters outside the concept. It is a two-dimensional plot of the 256 cluster centroids. The quantity plotted along each axis is defined by the user as either the intensity recorded in a single spectral band or some function of the intensities in two or more spectral bands. Here the horizontal coordinate ("brightness") and the vertical coordinate ("greenness") are linear transformations of the original seven spectral bands. That transformation is known to remote-sensing scientists as the "tasseled-cap" transformation. When the user selects a concept from the menu boxes in the scatter_plot window, all the points representing clusters that have been mapped to that concept are automatically highlighted in the assigned color. Here, for example, the user has selected the concept "agriculture," which is assigned a shade of bright green, so the points representing centroids whose clusters have been defined as part of the concept "agriculture" are colored bright green. The scatter_plot window reveals points close to those that are bright green and so represent other clusters whose brightness and greenness values fall near those already assigned to the concept but that have not themselves been assigned. Such clusters are also likely to represent agricultural fields, and the user can experiment by adding them to the concept and then examining the resulting mapping.

data stored for each pixel has been reduced from 7 bytes to 1 byte (the pixel's 1-byte cluster designation), a clustered image can be transmitted seven times faster than the original Landsat image. In addition, the great reduction in the quantity of the spectral data dramatically increases the speed of analysis. For example, calculating the "vegetation vigor" (which is a standard remote- sensing measure derived from the third and fourth spectral bands) of a 12-million-pixel-image from clustered data requires only 768 operations—three operations for each of the 256 clusters. Performing the same calculation on the original data requires 36 million operations, or three for each pixel.

Extracting concepts with Spectrum software. After the clustering algorithm reduces the dataset to a manageable size and level of detail, an interactive data-analysis tool called Spectrum is used to interpret the image. The Spectrum software package was developed through a collaboration between the Laboratory and the University of New Mexico. When using Spectrum, the human expert first identifies an example of a concept, and then the computer takes over and automatically identifies additional occurrences of that concept.

Figure 6 illustrates the interactive use of Spectrum on clustered Landsat data. The user simultaneously views the entire image as well as portions of it at two levels of magnification. To identify a concept of interest, the user simply draws a polygon enclosing a region of the image in one of the magnified views that corresponds, in his or her expert opinion, to that concept. This opinion may be based on the color or colors within the region, the intensity values in other bands of the spectrum, the region's shape, and/or the position

of the region with respect to other regions. For example, a region used to exemplify the concept "highway" would be gray in color, relatively warm in the thermal infrared band, narrow and relatively straight in shape, and might connect urban areas.

Once the user has drawn a polygon around a region that exemplifies a concept of interest, a category-labeling window appears on the screen. When the user enters the name of the concept, Spectrum automatically defines that concept as the set of all the spectral clusters that are associated with the pixels in the polygon. From the legend-control window, the user then selects a color to represent the concept. Spectrum will then automatically and instantly update the color of all pixels in the image that exemplify any of the spectral clusters that compose the concept. In the image shown in Figure 6, for example, the expert has identified and enclosed a portion of an agricultural region. Suppose that the region is composed of pixels that exemplify, or have been linked through clustering to, spectral clusters 20, 22, 36, and 48. As soon as the enclosed region is labeled "agriculture" and colored, say, bright green, clusters 20, 22, 36, and 48 are labeled "agriculture" and associated with the concept-specific shade of bright green. The color of every pixel in the entire image that exemplifies these clusters is then automatically updated to bright green. In other words, the concept "agriculture" is mapped onto the image.

After the computerized mapping is complete, the user can magnify and examine various regions to which the concept has been mapped and determine whether those regions do indeed exemplify the concept. Some fine tuning may be required. That is, some spectral clusters may be added or removed from the concept definition so that the mapping more accurately identifies examples of the concept.

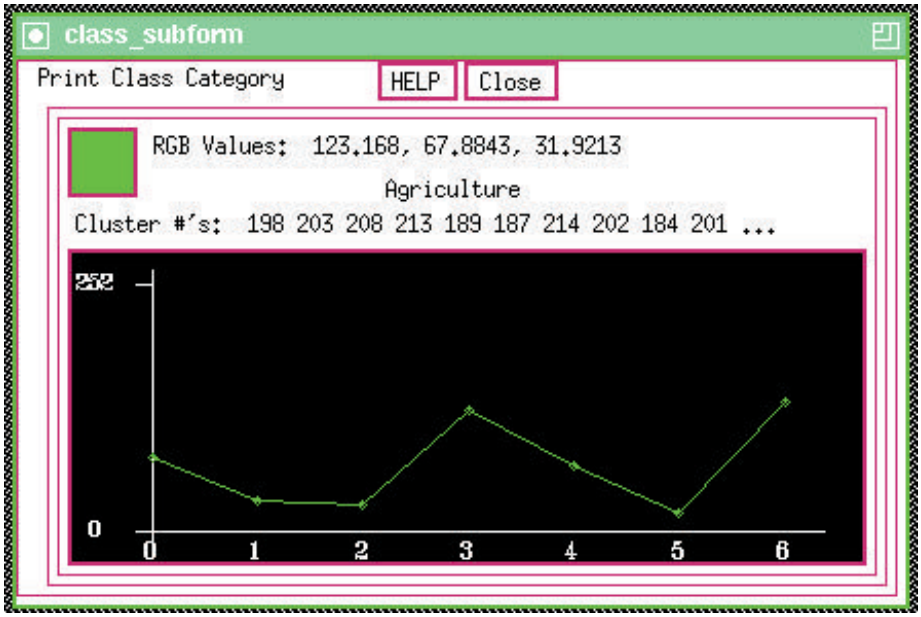


Figure 8. The Class_subform Window

This window plots the relative spectral intensities for any selected pixel. Users who are familiar with spectral data find that such a plot is a helpful addition to the visual information in the magnified images. Here, the spectral characteristics from a pixel in one of the bright green agricultural regions has been selected. Healthy vegetation is generally highly reflective in the infrared region of the spectrum. And, indeed, the plot shows that the infrared intensity measured for this pixel is relatively high. The user might gain added insights by comparing the shape of this plot to that produced by selecting pixels from other regions.

Sometimes the relevant concepts are more abstract and apply to regions with quite different spectral characteristics. For example, suppose an analyst is interested in determining what percentage of a given image is used for agriculture. To design the concept "agricultural field," the expert might draw two polygons, one around a field with growing crops and the other around a fallow field. The analyst would include both those regions in defining a single concept labeled "agriField" and both would be assigned a single color, say, dark blue. This concept unifies regions that are linked not by their spectral characteristics but rather by the more abstract idea of land use. When both fallow and planted fields identified as "agriFields," quantitative queries such as "what percentage of the image is used for agriculture?" can be posed and answered.

Figures 7 and 8 show two Spectrum windows used for displaying and comparing the spectral data associated with particular pixels, clusters, or concepts. The scatter_plot and class_subform windows provide the user with quantitative representations of the data that are useful tools for designing concepts and evaluating mapping results.

Application of the concept-extraction method to Landsat images has eliminated many of the problems associated with traditional analysis techniques.

The analyst need identify and label only one or a few examples of a concept, and the computer identifies all other examples. That division of labor, combined with the computational efficiency of clustered data, reduces the time required for analysis of an image from several weeks to two or three hours and also decreases the amount of time required to train new analysts. In addition, the data can be displayed on an inexpensive color screen. These advantages have sparked interest in our method from the United States Geological Survey, NASA, and the U.S. Army.

When applied to Landsat data, our concept-extraction technique has also proved capable of achieving the fundamental goal of data mining—finding human concepts amidst huge amounts of data. Obviously, the method would fail if the concepts were poorly separated. For example, our purpose would be subverted if we were to define "agriculture" to include clusters 17, 36, and 45, only to find many pixels of cluster 45 in the middle of the ocean. Such problems do arise occasionally, but the vast majority of cluster assignments are unambiguous. The key to our success lies in keeping clusters "fine-grained" by generating many more clusters than the number of concepts we want to consider.

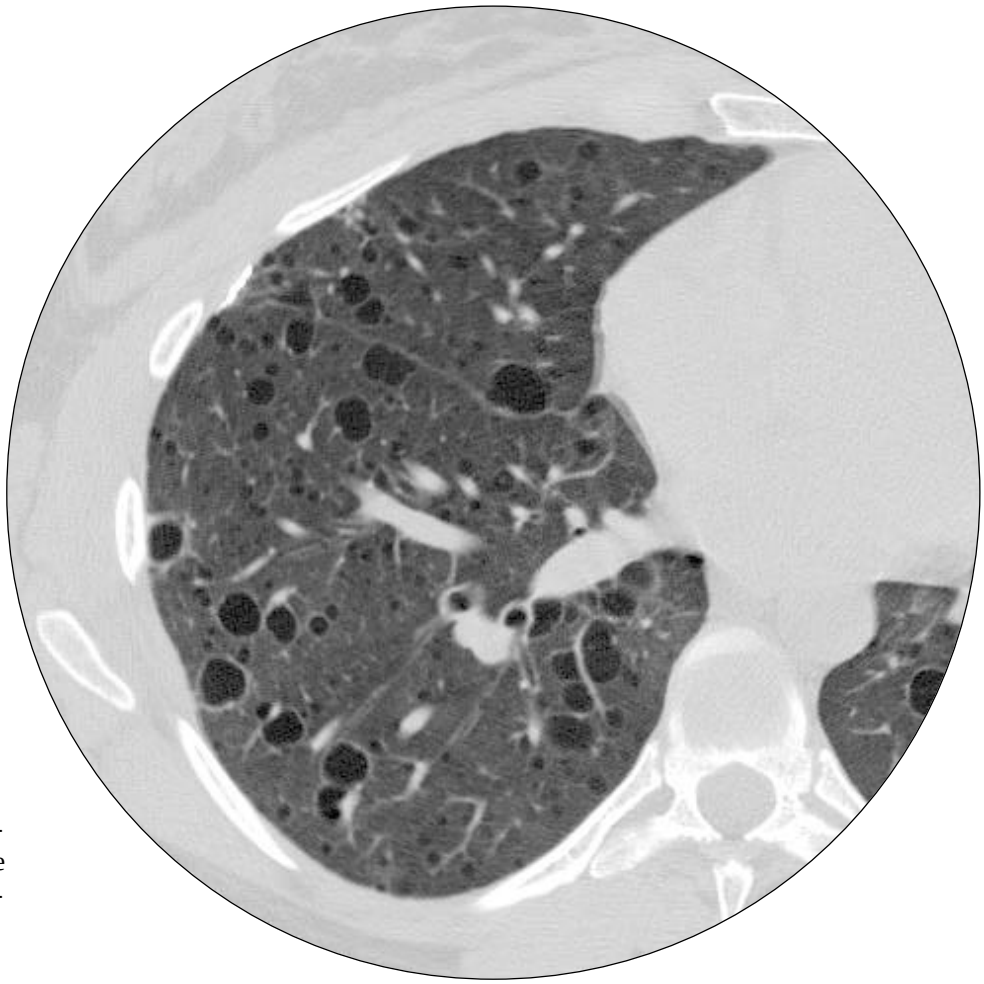
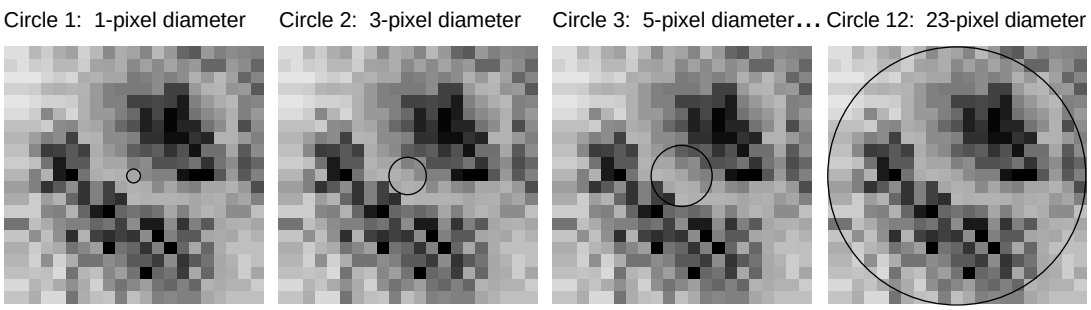


Figure 9. Computed-Tomography Lung Scan

Computed tomography is a method of recording two-dimensional x-ray images of an internal body structure. An incoming x-ray beam is absorbed by the lung tissue as well as the tissue and bone surrounding the lung. The intensities of the transmitted x rays are recorded by an array of charge counters analogous to those found in CCD cameras. Those intensities are recorded, processed, and reconstructed by a computer to form the image on a video display unit. The CT scan above shows a transverse slice of the left lung of a a woman afflicted with a moderate case of LAM disease. Both the spinal column (lower right) and the sternum (upper right) are visible in cross-sectional views. The scan shows multiple, thin-walled cysts throughout the lung. Several normal air-conducting bronchi are also visible, but they are difficult to distinguish from the cysts (see Figure 11). Both the large number of cysts and variations in cyst size make it difficult to gather quantitative diagnostic data on a routine basis.

Concept Extraction and CT Lung Scans

About LAM disease. In collaboration with Dr. John Newell at the National Jewish Center for Respiratory Illnesses in Denver, we are also applying our concept-extraction technique to the analysis of computed-tomography (CT or CAT) scans. Our effort has focused on the scans from women afflicted with lymphangioleiomyomatosis, or LAM disease (see Figure 9). This rare disease attacks only women in their childbearing years. The disease is characterized by cysts, or holes, in lung tissue. Its presence is signaled by profound shortness of breath, and diagnosis is confirmed by open-lung biopsy. The cause of the cysts is not yet fully understood. They may be a result of defective tissue that breaks down during normal breathing, or they may be indirectly caused by the proliferation of smooth muscle tissue within the lungs, a condition often seen in LAM patients. According to the latter theory, blockage



Circle number	Maximum intensity	Minimum intensity
1	247	247
2	247	150
3	768	125
.	.	.
.	.	.
.	.	.
12	3045	18

Figure 10. Creating a Quantitative Descriptor

The figure illustrates our method of preparing the CT data for clustering. Before clustering the data, we measure the maximum and minimum intensity of the gray values in twelve concentric circles of increasing diameter centered around each pixel. Only the first three circles and the last circle are illustrated here. The chart shows the maximum and minimum intensity values for each circle. The 24 intensity values are treated as a 24-component vector during clustering. As a result, each pixel is clustered not only in terms of its own gray-scale intensity value but also in terms of the values of the surrounding pixels. The 24-component vectors can be shown to vary according to the tissue density, texture, and local shape of structures in the area surrounding each pixel.

of the airways by the extra muscle tissue causes strain during breathing, which in turn tears the lung tissue. The current therapy involves hormonal manipulation, but unless a lung transplant is performed, LAM disease eventually leads to respiratory failure and death.

LAM disease is notoriously difficult to study. The amount of tissue that can safely be removed in a lung biopsy is too small for research purposes. Entire lungs can be studied when they are removed as part of a transplant operation or after an autopsy; however, the lungs collapse immediately upon removal, and the integrity of the tissue is compromised. Because of these difficulties, LAM researchers are turning to CT data to further their progress in studying this disease.

Previous studies done by "eyeballing" CT scans have indicated that as the disease progresses, patients show increasing numbers of large cysts in their lungs. Researchers are hoping that a more sophisticated analysis of the CT data will reveal more about the origin and progress of the disease and lead to improvements in diagnosis and treatment. We, and Dr. Newell, believe that an application of the concept-extraction technique may yield significant new results. In his words, concept-extracted CT data could provide a "quantitative, non-invasive, *in vivo* pathology."

Preparing CT data for concept extraction. Concept extraction is performed on the CT data by using a technique similar to that described above for the Landsat data. The CT data differ, however, in their initial representation and in how they are prepared for clustering. In the case of the Landsat data, "color," or intensity in seven bands of the electromagnetic spectrum, was the feature that allowed concepts to be differentiated from one another. In contrast, the CT scans record radiation intensity in only one spectral band, namely x rays, and so result in a single black-and-white digital image with pixel intensities ranging on a gray scale from 0 to 4095. These intensity variations alone do not provide enough information to differentiate cysts from normal bronchi because both cysts and bronchi register as "empty," or black, regions in the CT scan. (Bronchi are hollow regions and cysts are also hollow in the sense that they are holes or tears in the lung tissue.) If Spectrum were applied directly to the intensity data from the CT scan without any preprocessing, the method would fail. The user might draw a polygon within a cyst to create the concept "cyst." Spectrum would then identify and map as cysts all pixels with the same intensities as those in the polygon—that is, not only pixels composing additional cysts but also those that compose the bronchi.

To overcome this problem we have defined a quantitative descriptor characterizing the region surrounding each pixel. The descriptor consists of the maximum and minimum intensities of the pixels in each of 12 concentric circles of increasing diameter around each pixel, or 24 intensity measures per pixel (see Figure 10). This 24-component descriptor gives an indication of tissue density, texture, and local shape of structures surrounding a pixel. A special program has been developed to automatically scan the original data and record the 24 components of the descriptor for each pixel. The descriptors provide enough contextual information to avoid the complications outlined above and to produce accurate identification of cysts using Spectrum.

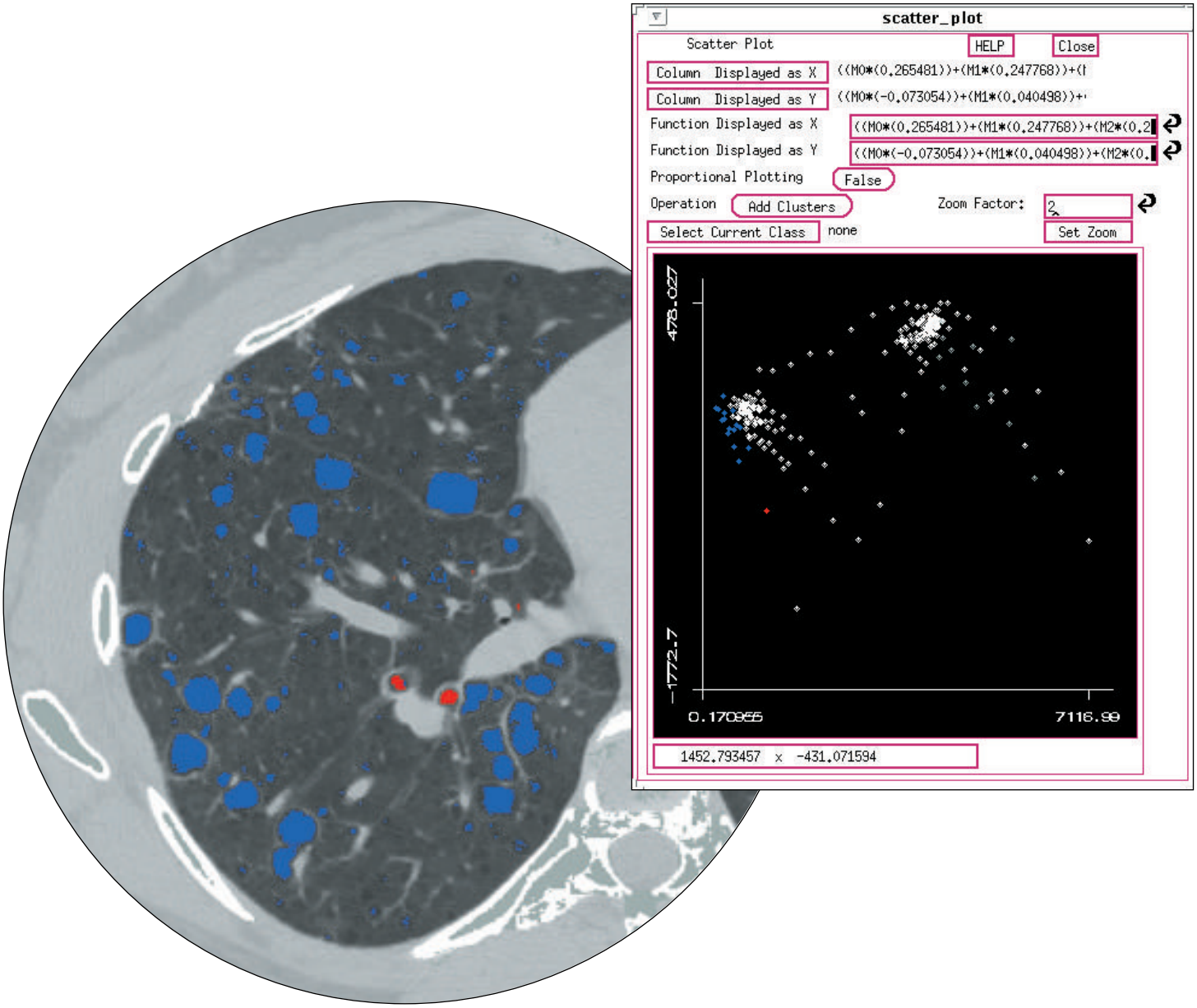


Figure 11. Application of Concept Extraction to a CT Lung Scan

At left is the scan shown in Figure 9 after application of the concept-extraction technique. The intensity data have been preprocessed by using the descriptor defined in Figure 10, and the resulting data have been clustered. Two concepts, "bronchus" and "cyst," have been defined for the clustered data by expert analysis. Spectrum has been used to map those concepts to the image; red represents "bronchus" and blue represents "cyst." Two large bronchi are visible near the center of the image, and the scan shows many cysts of various sizes. The descriptor derived from the nested set of circles described in Figure 10 is effective for distinguishing cysts from bronchi primarily because arteries run adjacent to and form a circular pattern around each bronchus. Once the concept-extraction method is applied to the clustered CT data to identify the cysts, researchers can easily carry out quantitative analysis of CT lung scans.

To the right of the CT scan is its scatter plot. The plot shows that the centroids (clusters) corresponding to the concept "cyst" (blue) are well separated from the centroids corresponding to the concept bronchus (red). The separation is achieved by using as the two axes for the plot the first and second principal components of the data: The *x*-axis is the first principal component, the direction in 24-dimensional space along which the data has maximal variance. The *y*-axis is the second principal component, the direction perpendicular to the first principal component along which the data shows maximal variance. Note that all the centroids corresponding to lung tissue fall toward the left in this plot, and the centroids corresponding to non-lung tissue fall toward the right. The centroids corresponding to cysts form a more-or-less diagonal line; those toward the upper left correspond to larger cysts and those toward the lower right correspond to smaller cysts.

Concept extraction and automated analysis of CT scans. Once the 24-component descriptors have been recorded for each pixel in a CT scan, concept extraction proceeds as outlined for the Landsat data. The continuous *k*-means algorithm is used to partition the descriptors into 256 clusters. Spectrum is then used to define four concepts—cyst, bronchus, normal lung tissue, and non-lung material—and to map them to the CT image. Figure 11 shows the lung scan shown in Figure 9 after the first two of these concepts have been mapped to that image. Cysts are shown in blue and bronchi in red. Figure 11 also shows a scatter plot for this image that illustrates the clear separation between the concepts "bronchus" and "cyst" and thus the success of clustering based on the 24-component "contextual" descriptor.

To automate the analysis of concept-mapped data, Los Alamos researchers developed a program that counts the number of cysts of different sizes. This program uses a region-growing algorithm in which a cyst is defined as a group of adjacent pixels each of which has been identified by Spectrum as belonging to the concept "cyst." The size of the cyst is determined by the number of such adjacent pixels along with the pixel resolution of the CT imagery. The program can also quantify, by means of a central-moment method, other descriptive features of the cyst including eccentricity (deviation from a circular shape) and orientation.

Although our results are preliminary—we have not yet systematically examined a large number of CT scans—the concept-extraction method promises to reveal useful new information about LAM disease. The scans we have analyzed to date support the earlier evidence that more large cysts appear in the later stages of the disease. In addition, computer-assisted techniques have revealed large numbers of small cysts in the later stages of the disease. In other words, not only do small cysts grow larger as the disease progresses but also many new cysts develop. Figure 12 shows graphs of the frequency of different-sized cysts in scans of patients in two stages of the disease. As expected, the patient in the more advanced stage has more large cysts than the patient in the earlier stage; she also shows a large number of new small cysts. To confirm or refute the proliferation of small cysts in association with the later stages of the disease, the analysis must be repeated on a more substantial number of CT scans.

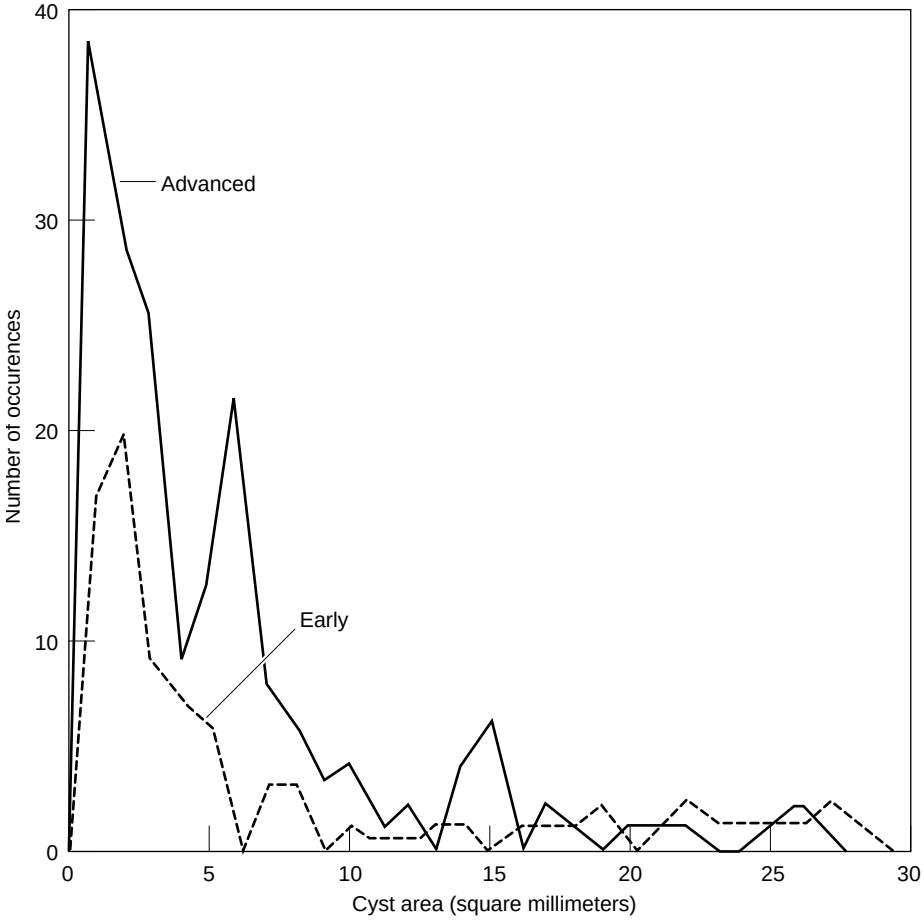


Figure 12. Frequency Distribution of Cyst Sizes

The graphs show the number of cysts of various sizes for two LAM-disease patients. The dotted line represents data from a patient early in the course of the disease, and the solid line, a patient in the later stages. This comparative plot shows that the patient in the later stage not only has more large cysts but also has roughly twice as many small cysts as the patient in the early stage. These results suggest that as the disease progresses, small cysts grow in size and many new small cysts appear. Before the availability of the concept-extraction technique described in the main text, researchers had no efficient way to obtain quantitative measurements of cyst size and number. Now, the progress of the disease in individual patients can be monitored, and researchers can also gain meaningful insights by comparing data from many patients.

continued on page 145

Clustering and the Continuous *k*-Means Algorithm

Vance Faber

Many types of data analysis, such as the interpretation of Landsat images discussed in the accompanying article, involve datasets so large that their direct manipulation is impractical. Some method of data compression or consolidation must first be applied to reduce the size of the dataset without losing the essential character of the data. All consolidation methods sacrifice some detail; the most desirable methods are computationally efficient and yield results that are—at least for practical applications—representative of the original data. Here we introduce several widely used algorithms that consolidate data by clustering, or grouping, and then present a new method, the continuous *k*-means algorithm,* developed at the Laboratory specifically for clustering large datasets.

Clustering involves dividing a set of data points into non-overlapping groups, or clusters, of points, where points in a cluster are "more similar" to one another than to points in other clusters. The term "more similar," when applied to clustered points, usually means closer by some measure of proximity. When a dataset is clustered, every point is assigned to some cluster, and every cluster can be characterized by a single reference point, usually an average of the points in the cluster. Any particular division of all points in a dataset into clusters is called a partitioning.

One of the most familiar applications of clustering is the classification of plants or animals into distinct groups or species. However, the main purpose of clustering Landsat data is to reduce the size and complexity of the dataset. Data reduction is accomplished by replacing the coordinates of each point in a cluster with the coordinates of that cluster's reference point. Clustered data require considerably less storage space and can be manipulated more quickly than the original data. The value of a particular clustering method will depend on how closely the reference points represent the data as well as how fast the program runs.

A common example of clustering is the consolidation of a set of students' test scores, expressed as percentages, into five clusters, one for each letter grade A, B, C, D, and F (see Figure 1). The test scores are the data points, and each cluster's reference point is the average of the test scores in that cluster. The letter grades can be thought of as symbolic replacements for the numerical reference points.

Test scores are an example of one-dimensional data; each data point represents a single measured quantity. Multidimensional data can include any number of measurable attributes; a biologist might use four attributes of duck bills (four-dimensional data: size, straightness, thickness, and color) to sort a large set of ducks into several species. Each independent characteristic, or measurement, is one dimension. The consolidation of large, multidimensional datasets is the main pur-

* The continuous *k*-means algorithm is part of a patented application for improving both the processing speed and the appearance of color video displays. The application is commercially available for Macintosh computers under the names *Fast Eddie*, ©1992 and *Planet Color*, ©1993, by Paradigm Concepts, Inc., Santa Fe, NM. This software was developed by Vance Faber, Mark O. Mundt, Jeffrey S. Saltzman, and James M. White.

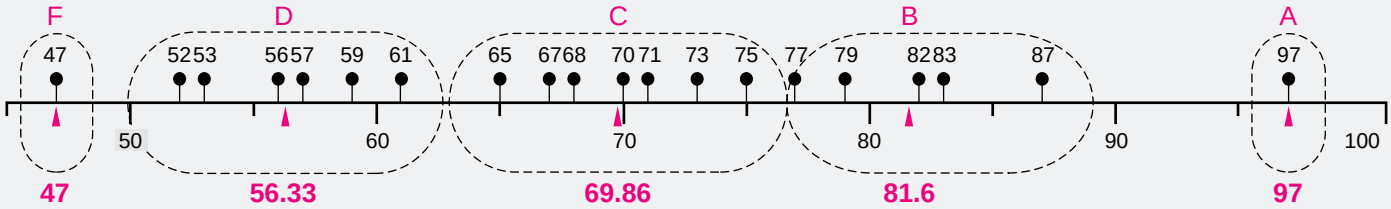


Figure 1. Clustering Test Scores
The figure illustrates an arbitrary partitioning of 20 test scores into 5 non-overlapping clusters (dashed lines), corresponding to 5 letter grades. The reference points (means) are indicated in red.

pose of the field of cluster analysis. We will describe several clustering methods below. In all of these methods the desired number of clusters k is specified beforehand. The reference point z_i for the cluster i is usually the centroid of the cluster. In the case of one-dimensional data, such as the test scores, the centroid is the arithmetic average of the values of the points in a cluster. For multidimensional data, where each data point has several components, the centroid will have the same number of components and each component will be the arithmetic average of the corresponding components of all the data points in the cluster.

Perhaps the simplest and oldest automated clustering method is to combine data points into clusters in a pairwise fashion until the points have been condensed into the desired number of clusters; this type of agglomerative algorithm is found in many off-the-shelf statistics packages. Figure 2 illustrates the method applied to the set of test scores given in Figure 1.

There are two major drawbacks to this algorithm. First—and absolutely prohibitive for the analysis of large datasets—the method is computationally inefficient. Each step of the procedure requires calculation of the distance between every possible pair of data points and comparison of all the distances. The second difficulty is connected to a more fundamental problem in cluster analysis: Although the algorithm will always produce the desired number of clusters, the centroids of these clusters may not be particularly representative of the data.

What determines a "good," or representative, clustering? Consider a single cluster of points along with its centroid or mean. If the data points are tightly clustered around the centroid, the centroid will be representative of all the points in that cluster. The standard measure of the spread of a group of points about its mean is the variance, or the sum of the squares of the distance between each point and the mean. If the data points are close to the mean, the variance will be small. A generalization of the variance, in which the centroid is replaced by a reference point that may or may not be a centroid, is used in cluster analysis to indicate the overall quality of a partitioning; specifically, the error measure E is the sum of all the variances:

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \| x_{ij} - z_i \|^2,$$

where x_{ij} is the jth point in the ith cluster, z_i is the reference point of the ith cluster, and n_i is the number of points in that cluster. The notation $\|x_{ij} - z_i\|$ stands for the distance between x_{ij} and z_i. Hence, the error measure E indicates the overall spread of data points about their reference points. To achieve a representative clustering, E should be as small as possible.

The error measure provides an objective method for comparing partitionings as well as a test for eliminating unsuitable partitionings. At present, finding the best

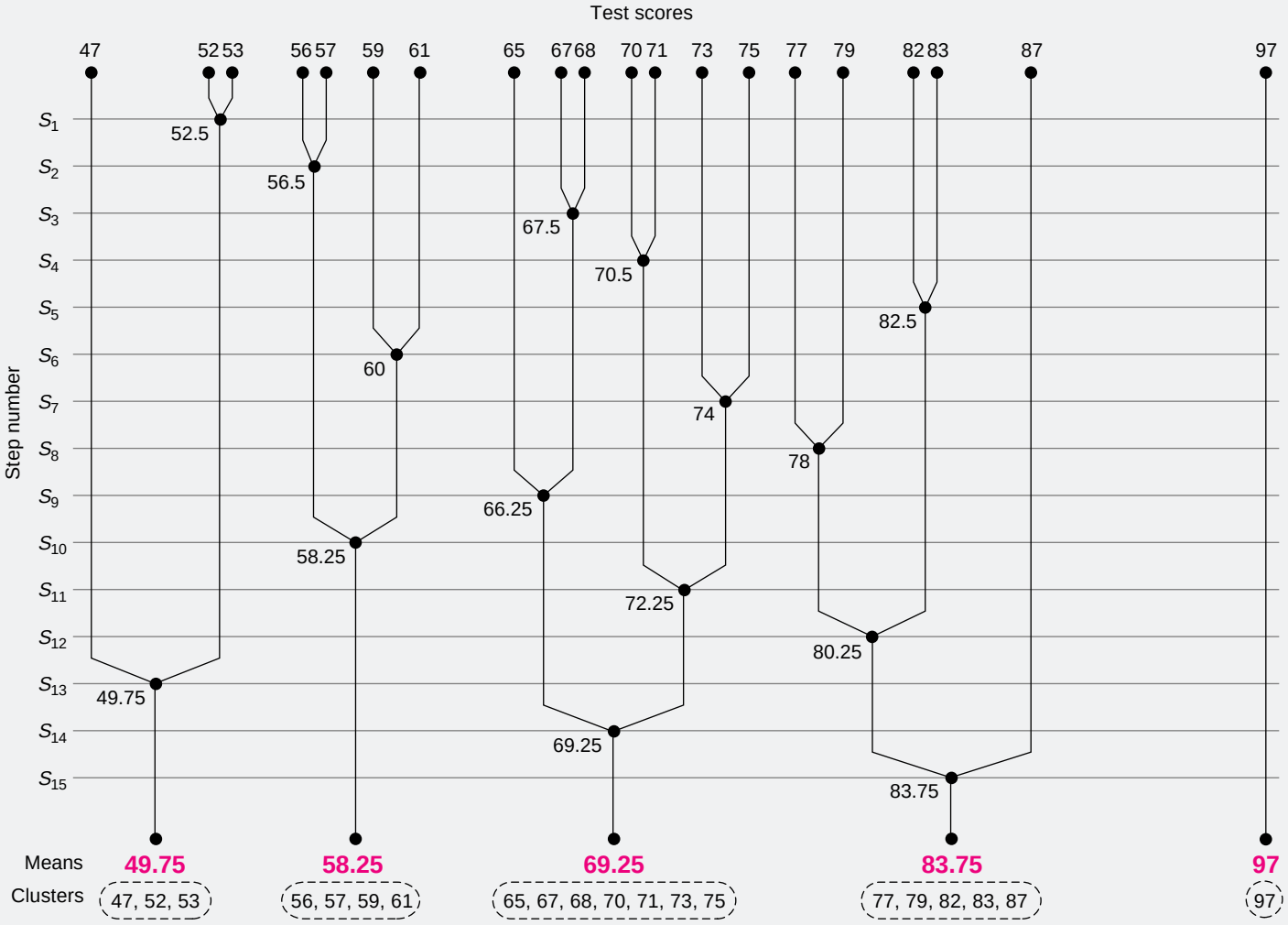


Figure 2. Pairwise Agglomerative Clustering

The figure illustrates the operation of an agglomerative clustering method, in which the 20 test scores of Figure 1 are successively merged by pairs of points and/or pairs of clusters until all the scores are collected into 5 clusters. The steps of the algorithm are shown in the branching of a dendrogram, or tree structure (much like a genealogy). A node, or branch point, indicates the merging of two branches into one, i.e. two data points into one cluster, or two clusters into one larger cluster. The algorithm begins with 20 separate clusters of one point apiece. For the first step in the algorithm, the closest two points (here, scores of 52 and 53) are found and merged into one cluster {52,53}. The two individual points are replaced by a single point equal to the unweighted average of the two points (52.5). The next step repeats this process (find the closest two points, calculate the average, merge the points), but with 19 points and 19 clusters (18 one-point clusters, plus 1 two-point cluster). There will be only one new branch, or merge at each step. Hence, if there is more than one pair of points at the minimum distance, only one pair will be merged at each step. It takes 15 steps to consolidate 20 points into 5 clusters.

partitioning (the clustering most representative of an arbitrary dataset) requires generating all possible combinations of clusters and comparing their error measures. This can be done for small datasets with a few dozen points, but not for large sets—the number of different ways to combine 1 million data points into 256 clusters, for example, is $256^{1,000,000}/256!$, where 256! is equal to $256 \times 255 \times 254 \times \cdots \times 2 \times 1$. This number is greater than $10^{2,000,000}$, or 1 followed by 2 million zeros.

When clustering is done for the purpose of data reduction, as in the case of the Landsat images, the goal is not to find the best partitioning. We merely want a reasonable consolidation of N data points into k clusters, and, if necessary, some efficient way to improve the quality of the initial partitioning. For that purpose, there is a family of iterative-partitioning algorithms that is far superior to the agglomerative algorithm described above.

Iterative algorithms begin with a set of k reference points whose initial values are usually chosen by the user. First, the data points are partitioned into k clusters: A data point x becomes a member of cluster i if z_i is the reference point closest to x. The positions of the reference points and the assignment of the data points to clusters are then adjusted during successive iterations. Iterative algorithms are thus similar to fitting routines, which begin with an initial "guess" for each fitted parameter and then optimize their values. Algorithms within this family differ in the details of generating and adjusting the partitions. Three members of this family are discussed here: Lloyd's algorithm, the standard *k*-means algorithm, and a continuous *k*-means algorithm first described in 1967 by J. MacQueen and recently developed for general use at Los Alamos.

Conceptually, Lloyd's algorithm is the simplest. The initial partitioning is set up as described above: All the data points are partitioned into k clusters by assigning each point to the cluster of the closest reference point. Adjustments are made by calculating the centroid for each of those clusters and then using those centroids as reference points for the next partitioning of all the data points. It can be proved that a local minimum of the error measure E corresponds to a "centroidal Voronoi" configuration, where each data point is closer to the reference point of its cluster than to any other reference point, and each reference point is the centroid of its cluster. The purpose of the iteration is to move the partition closer to this configuration and thus to approach a local minimum for E.

For Lloyd's and other iterative algorithms, improvement of the partitioning and convergence of the error measure E to a local minimum is often quite fast—even when the initial reference points are badly chosen. However, unlike guesses for parameters in simple fitting routines, slightly different initial partitionings generally do not produce the same set of final clusters. A final partitioning will be better than the initial choice, but it will not necessarily be the best possible partitioning. For many applications, this is not a significant problem. For example, the differences between Landsat images made from the original data and those made from the clustered data are seldom visible even to trained analysts, so small differences in the clustered data are even less important. In such cases, the judgment of the analyst is the best guide as to whether a clustering method yields reasonable results.

(a) Setup:
Reference point 1 (filled red circle) and reference point 2 (filled black circle) are chosen arbitrarily. All data points (open circles) are then partitioned into two clusters: each data point is assigned to cluster 1 or cluster 2, depending on whether the data point is closer to reference point 1 or 2, respectively.

(b) Results of first iteration:
Next each reference point is moved to the centroid of its cluster. Then each data point is considered in the sequence shown. If the reference point closest to the data point belongs to the other cluster, the data point is reassigned to that other cluster, and both cluster centroids are recomputed.

(c) Results of second iteration:
During the second iteration, the process in Figure 3(b) is performed again for every data point. The partition shown above is stable; it will not change for any further iteration.

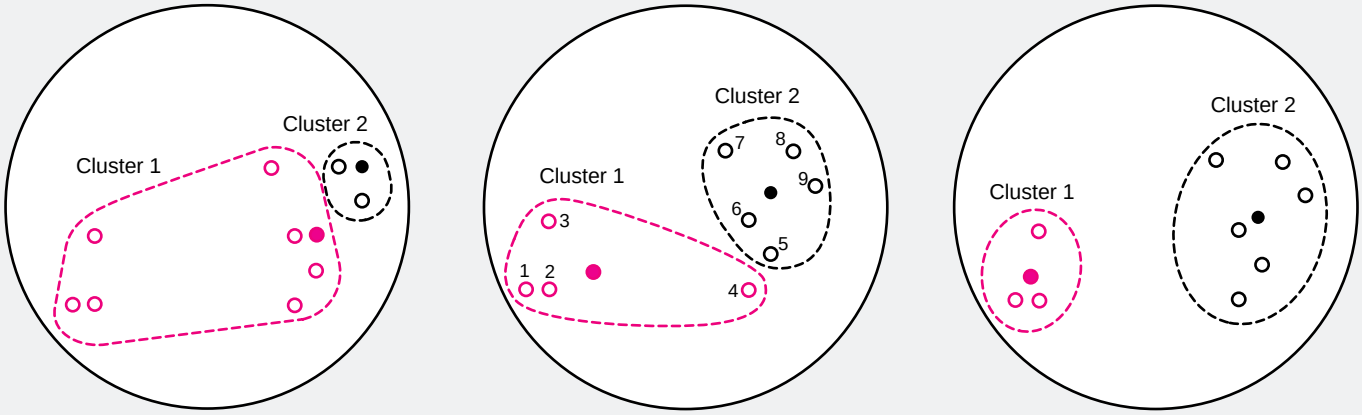


Figure 3. Clustering by the Standard *k*-Means Algorithm
The diagrams show results during two iterations in the partitioning of nine two-dimensional data points into two well-separated clusters, using the standard *k*-means algorithm. Points in cluster 1 are shown in red, points in cluster 2 are shown in black; data points are denoted by open circles and reference points by filled circles. Clusters are indicated by dashed lines. Note that the iteration converges quickly to the correct clustering, even for this bad initial choice of the two reference points.

The standard *k*-means algorithm differs from Lloyd's in its more efficient use of information at every step. The setup for both algorithms is the same: Reference points are chosen and all the data points are assigned to clusters. As with Lloyd's, the *k*-means algorithm then uses the cluster centroids as reference points in subsequent partitionings—but the centroids are adjusted both during and after each partitioning. For data point x in cluster i, if the centroid z_i is the nearest reference point, no adjustments are made and the algorithm proceeds to the next data point. However, if the centroid z_j of the cluster j is the reference point closest to data point x, then x is reassigned to cluster j, the centroids of the "losing" cluster i (minus point x) and the "gaining" cluster j (plus point x) are recomputed, and the reference points z_i and z_j are moved to their new centroids. After each step, every one of the k reference points is a centroid, or mean, hence the name "*k*-means." An example of clustering using the standard *k*-mean algorithm is shown in Figure 3.

There are a number of variants of the *k*-means algorithm. In some versions, the error measure E is evaluated at each step, and a data point is reassigned to a different cluster only if that reassignment decreases E. In MacQueen's original paper on the *k*-means method, the centroid update (assign data point to cluster, recompute the centroid, move the reference point to the centroid) is applied at each step in the initial partitioning, as well as during the iterations. In all of these cases, the standard *k*-means algorithm requires about the same amount of computation for a single pass through all the data points, or one iteration, as does Lloyd's algorithm. However, the *k*-means algorithm, because it constantly updates the clusters, is unlikely to require as many iterations as the less efficient Lloyd's algorithm and is therefore considerably faster.

The Continuous *k*-Means Algorithm

The continuous *k*-means algorithm is faster than the standard version and thus extends the size of the datasets that can be clustered. It differs from the standard version in how the initial reference points are chosen and how data points are selected for the updating process.

In the standard algorithm the initial reference points are chosen more or less arbitrarily. In the continuous algorithm reference points are chosen as a random sample from the whole population of data points. If the sample is sufficiently large, the distribution of these initial reference points should reflect the distribution of points in the entire set. If the whole set of points is densest in Region 7, for example, then the sample should also be densest in Region 7. When this process is applied to Landsat data, it effectively puts more cluster centroids (and the best color resolution) where there are more data points.

Another difference between the standard and continuous *k*-means algorithms is the way the data points are treated. During each complete iteration, the standard algorithm examines all the data points in sequence. In contrast, the continuous algorithm examines only a random sample of data points. If the dataset is very large and the sample is representative of the dataset, the algorithm should converge much more quickly than an algorithm that examines every point in sequence. In fact, the continuous algorithm adopts MacQueen's method of updating the centroids during the initial partitioning, when the data points are first assigned to clusters. Convergence is usually fast enough so that a second pass through the data points is not needed.

From a theoretical perspective, random sampling represents a return to MacQueen's original concept of the algorithm as a method of clustering data over a continuous space. In his formulation, the error measure E_i for each region R_i is given by

$$E_i = \int_{x \in R_i} \rho(x) \, \| x - z_i \|^2 dx \, ,$$

where $\rho(x)$ is the probability density function, a continuous function defined over the space, and the total error measure E is given by the sum of the E_i's. In MacQueen's concept of the algorithm, a very large set of discrete data points can be thought of as a large sample—and thus a good estimate—of the continuous probability density $\rho(x)$. It then becomes apparent that a random sample of the dataset can also be a good estimate of $\rho(x)$. Such a sample yields a representative set of cluster centroids and a reasonable estimate of the error measure without using all the points in the original dataset.

These modifications to the standard algorithm greatly accelerate the clustering process. Since both the reference points and the data points for the updates are chosen by random sampling, more reference points will be found in the densest regions of the dataset and the reference points will be updated by data points in the most critical regions. In addition, the initial reference points are already members of the dataset and, as such, require fewer updates. Therefore, even when applied to a large dataset, the algorithm normally converges to a solution after only a small fraction (10 to 15 percent) of the total points have been examined. This rapid convergence distinguishes the continuous *k*-means from less efficient algorithms. Clustering with the continuous *k*-means algorithm is about ten times faster than clustering with Lloyd's algorithm.

The computer time can be further reduced by making the individual steps in the algorithm more efficient. A substantial fraction of the computation time required by any of these clustering algorithms is typically spent in finding the reference point closest to a particular data point. In a "brute-force" method, the distances from a given data point to all of the reference points must be calculated and compared. More elegant methods of "point location" avoid much of this time-consuming process by reducing the number of reference points that must be considered—but some computational time must be spent to create data structures. Such structures range from particular orderings of reference points, to "trees" in which reference points are organized into categories. A tree structure allows one to eliminate entire categories of reference points from the distance calculations. The continuous *k*-means algorithm uses a tree method to cluster three-dimensional data, such as pixel colors on a video screen. When applied to seven-dimensional Landsat data, the algorithm uses single-axis boundarizing, which orders the reference points along the direction of maximum variation. In either method only a few points need be considered when calculating and comparing distances. The choice of a particular method will depend on the number of dimensions of the dataset.

Two features of the continuous *k*-means algorithm—convergence to a feasible group of reference points after very few updates and greatly reduced computer time per update—are highly desirable for any clustering algorithm. In fact, such features are crucial for consolidating and analyzing very large datasets such as those discussed in the accompanying article. □

Further Reading

James M. White, Vance Faber, and Jeffrey S. Saltzman. 1992. Digital color representation. U.S. Patent Number 5,130,701.

Stuart P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* IT-28: 129–137.

Edward Forgy. 1965. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications, *Biometrics* 21: 768.

J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I, Statistics.* Edited by Lucien M. Le Cam and Jerzy Neyman. University of California Press.

Jerome H. Friedman, Forest Baskett, and Leonard J. Shustek. 1975. An algorithm for finding nearest neighbors. *IEEE Transactions on Computers* C-24: 1000-1006. [Single-axis boundarizing, dimensionality.]

Jerome H. Friedman Jon Louis Bentley, and Raphael Ari Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software* 3: 209–226. [Tree methods.]

Helmuth Späth. 1980. *Cluster Analysis Algorithms for Data Reduction and Classification of Objects.* Halsted Press.

Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for Clustering Data*. Prentice Hall.

The biography of Vance Faber appears on page 149.

Concept Extraction Applied to Text Analysis of Medical Records

Challenges in analyzing textual data. Textual datasets are very different from the digital-image data to which we originally applied our concept-extraction technique. Since the technique had been conceived in a general way and was not specific to any particular kind of data, we felt confident that our method could also be applied in the analysis of textual data. Our attempts are in their early stages and several details still need to be worked out, but our progress to date has served to reinforce our confidence in the approach. The textual data consist of transcripts of physicians' notes on patient visits to a large health-maintenance organization (HMO). Each transcription is termed a document, and each document is a unique data element. The sample we analyzed contains 142,475 such documents—the entire HMO output for a recent ten-month period. To ensure privacy, the patients' names and medical-record numbers are replaced by randomly chosen pseudonyms, access to the data is strictly controlled, and all identifying information is altered in publicly distributed documents.

Current methods of analysis. The current method of extracting information from transcribed patient records is to bind printed copies of all documents related to a single patient in a medical record folder and deliver that folder to the physician whenever the patient is treated at the HMO. The physician then reviews the documents and extracts information deemed helpful for determining a course of treatment. On occasion, epidemiologists or medical researchers will collect a small sample of related cases and study the patients' records in an attempt to answer general medical questions related to treatments and outcomes.

Obviously, these two extraction methods are limited by the ability of highly-trained human readers to accurately and comprehensively read the documents. A dataset of over 100,000 such documents cannot be examined with current methods, yet the document set as a whole probably contains a considerable amount of useful information, such as answers to questions like: What diseases are most common in this patient population? Is aspirin an effective treatment for cystitis? Is the rate of attempted suicide higher among teenagers or adults? Answering such questions requires a computerized method of examining large datasets of textual documents and extracting conceptual information.

It may seem that the solution is not to train computers to analyze text but rather to train physicians to record their patient encounters in a structured way so it will be easier to analyze the data—"fill out the form!" Many attempts have been made to standardize medical-data recording, but all have been successfully resisted by physicians. They argue that only free text can adequately capture the inherent ambiguity of the concepts involved in a medical encounter. Consider the concept "pain." Pain may be constant or intermittent or associated with a specific movement, and there are many different ways of describing it: sharp, dull aching, throbbing, and so on. Physicians are keenly aware of the inherent ambiguity of conceptual labels and have insisted on maintaining their freedom to record encounters in a free-text format. Therefore, if the information in the dataset is to be made more generally useful, we must solve the difficult problem of extracting it from documents written in a free-text format.

Choosing a quantitative descriptor and measure of "closeness." Before the dataset can be clustered, we must choose a quantitative descriptor for each data element. The most obvious numerical representation of text is a descriptor indicating the frequencies of all words in each document. However, since the number of words encountered in medical records is very large (hundreds of thousands) and is constantly increasing, we wanted to avoid the problem of managing such huge descriptors. Choosing a limited set of words based on expert recommendations might reduce the difficulties, but even an expert might overlook terms of great significance. In addition, parsing ASCII text into "words" presents problems with respect to handling compound words, abbreviations, homonyms, hyphenations, spelling errors, morphological alternatives (such as *nausea* and *nauseous*), and so on.

Our solution to these problems is the use of a quantitative representation of each document that is based on words but avoids using the words themselves. The method, called character-trigram representation, represents each document as the set of frequencies of all three-letter sequences (trigrams) in the document. To create this representation, all non-alphabetic characters in the document, including spaces, are removed and all letters are reduced to lower case. A three-letter window is then run over the document and the frequencies of the different trigrams is tallied. The phrase "Prozac 20 mg daily," for example, yields one instance each of trigrams *pro*, *roz*, *oza*, *zac*, *acm*, *cmg*, *mgd*, *gda*, *dai*, *ail*, and *ily*. Trigrams—or more generally, *n*-grams of varying lengths—have been used successfully in text applications ranging from spell checking to language identification.

The main advantage of the trigram approach is that the number of trigrams

continued from page 137

that must be tallied is relatively small—about 14,000. [The total number of trigrams that can be found in words written in the Latin alphabet is 17,576 (26^3), but many of the possible trigrams, including, for example, *bbb*, and *yzv*, are seldom if ever encountered.] So the use of trigrams brings the clustering problem down to a size that our continuous *k*-means algorithm can easily handle without our having to select a list of words beforehand. In addition, trigrams can be tallied without any of the difficult preprocessing that a tally of words would require. Variations in word form due to spelling errors and morphological variations "wash out" because trigrams provide a shared representation for word variants. For example, morphological variants such as congested and congestion overlap and share the trigrams *con*, *ong*, *nge*, *ges*, and *est*. Likewise, *telephone* and *telepbone* (the latter the result of a plausible optical-character-recognition error) share trigrams *tel*, *ele*, *lep*, and *one*.

The next task before clustering is the selection of a quantitative measure of the similarity, or "closeness," of two sets of trigram frequencies—which provides a measure of the "closeness" of two documents. We chose to consider the sets as vectors in a space of about 14,000 dimensions, one dimension for each trigram, and to measure the distance in terms of the cosine of the angle between the trigram vectors for the two documents. The angle between two vectors does not depend on their lengths, so this measure allows the unbiased comparison of documents of different lengths. We calculate the cosine according to a standard formula of vector analysis. If the cosine is equal to 1 (meaning the vectors are collinear), the documents have identical trigram-frequency distributions and are very similar if not identical. If the cosine is 0 (meaning the vectors are perpendicular), then the documents have no trigrams in common, and thus no words in common. More complicated measures, which take into account the relative variability of specific trigrams across the dataset, could be constructed and may prove useful. An obvious enhancement would be to weight common trigrams such as *the* and *and* less heavily than others. In our early stage of exploration, however, we have chosen the simplest approach.

Testing the method. Before attempting to cluster the documents, we wanted to be certain that documents that are close to one another according to our cosine measure are also close to one another in the sense that they relate to the same topic. To test our method we selected a document that was clearly about headaches and searched the dataset for the closest, or most similar, documents according to the trigram-frequency descriptor. Of the ten closest documents, eight related to headaches, one to a numb foot, and one to dizziness. Those documents that were not related to headaches were, however, generated by the same doctor and written on the same day as the document relating to headaches selected as our reference document.

These results were compared with results obtained by using a popular text retrieval tool (WAIS), which works by counting the number of words in common between two documents. We found that WAIS returned one document about headaches, the two non-headache documents our method also found, and seven other documents that were about a variety of medical topics including stroke, facial tic, and back pain. Given this favorable comparison, we proceeded with clustering under the assumption that similarity in "trigram space" indicates similarity in "topic space." It is important to note that the documents have different meanings—each describes very different situations. The similarity resides in the topic of the documents. Since they contain similar root words describing that topic, they produce similar trigram distributions. The sentences "she is healthy" and "he is unhealthy" have different meanings, but both are about the same topic—health—and they share a similar trigram distribution.

Clustering the documents. We chose to create 1000 clusters, and the results yielded various types of clusters. One cluster is very large, containing over 900 documents. All members of this cluster appear to be documents that concern vague headaches. Without any further analysis, we can reasonably conclude that headaches are a very common symptom prompting numerous visits to the HMO; visits relating to headaches should be considered seriously when planning resource allocations. It is very likely, however, that other clusters also relate to headaches, and it is necessary to fuse such clusters into a single conceptual category before meaningful quantitative estimates can be made. It is important to note, however, that the clustering process produces valuable results even without fusing clusters. Clustering of a very large dataset greatly reduces the time needed to locate similar documents, since one can first find the cluster to which the reference document belongs and then search only the members of that and nearby clusters for similar documents. This procedure is much faster than using an algorithm that considers all documents as candidates, but it yields virtually identical results.

Our results yielded a second type of cluster, a singleton, which contains only one document. Any document found in a singleton usually relates to some unique topic or is simply some sort of

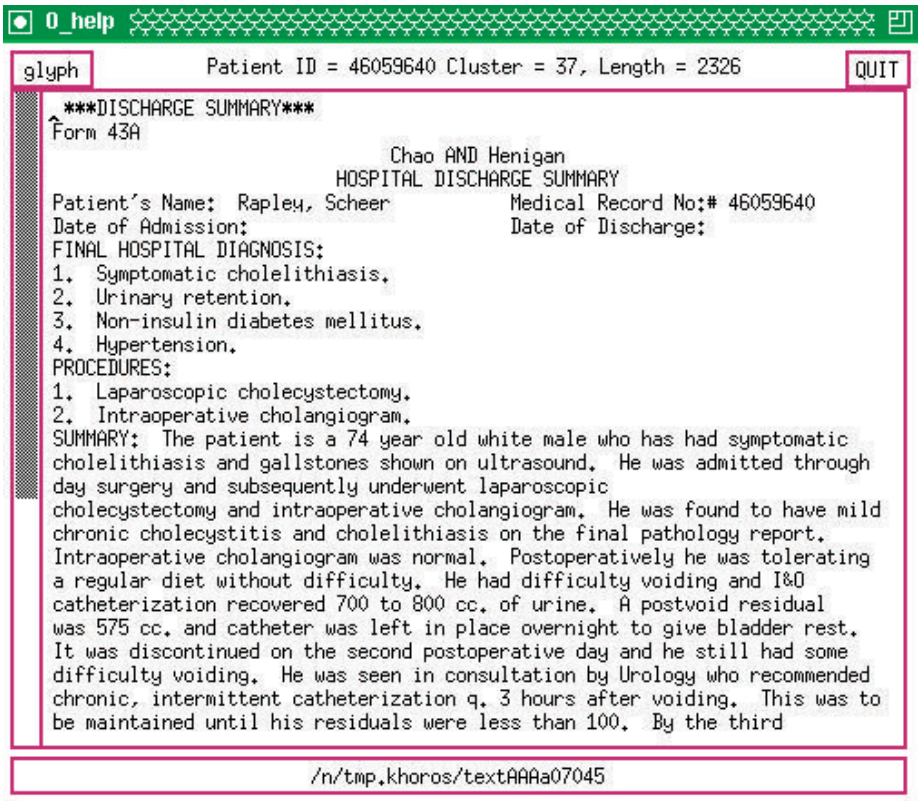


Figure 13. Example of a Two - Member Cluster

The figure shows an example of the type of document (transcripts from doctors' notes on-patient visits) we have been working with in our textual dataset. These two documents represent a doubleton cluster, a cluster containing only two documents that are so much more similar to each other than to the other documents that the algorithm automatically separated them into their own cluster. Here, both documents relate to cases of laparascopic cholestectomy, both patients are white males in their mid-70s, and the documents share several other medical details in common. Since laparascopic cholestectomy is a fairly common procedure, it seems likely that there are more instances of this procedure in other clusters.

error in the transcription process. A third type of cluster is the doubleton, which contains two very similar documents (see Figure 13).

Fusing related clusters. Any closely related clusters must be fused before any quantitative information about a given condition, procedure, or therapy can be obtained, and we are currently trying to develop an efficient method of fusing such clusters. The underlying difficulty is that people are best able to evaluate the clusters based on words, but the clusters were created from their trigram content. We are making some progress in finding a way to map the trigram distributions back to a meaningful list of words so that the expert can fuse seemingly disjoint clusters into conceptually relevant categories.

The basic idea is to rank the words in each document according to how closely they mirror the cluster centroid. Words that contain trigrams that occur frequently in the cluster centroid are considered to be more indicative of that centroid than those containing less frequently occurring trigrams. To begin, we assign a weight for each possible trigram by taking the normalized frequency of each trigram in the cluster centroid. We then determine the importance of each letter in every document in the cluster by adding the weights of the three trigrams to which that letter belongs, given its left and right context. A word-weight is constructed by taking the average letter-weight for the entire word. A ranked word-list can then be compiled. A "stop" list is applied to eliminate non-content words, and a stemming algorithm is applied so that redundancy in the ranked list is reduced by eliminating multiple occurrences of words that are simple variants of the same stem (*operate*, *operates*, *operating*, etc.) Finally, a list of the ten most important (that is, the ten most heavily weighted) words is created for each cluster.

The top ten members of this list for the two-member cluster in Figure 13 are: *tolerate*, *operate*, *intake*, *postoperative*, *lifting*, *tolinase*, *white*, *intraoperative*, *laparoscopic*, and *stone*. This list catalogues keywords describing the most important elements of the centroid for this cluster. Such lists are useful for retrieving documents on the basis of keyword searches. The lists also represent practical, quick-reference descriptions of the contents of the various clusters, and the user can decide whether to read particular documents by examining these lists.

We are also planning a user-interface to help the expert decide which clusters should be fused. It is impractical to expect an expert to compare and evaluate all the lists for the 1000 clusters—even glancing at such lists is an onerous task. We plan to provide an interface that presents a small group of similar lists (similar according to a measure of the co-occurrence of either trigrams or words), prompts the expert to decide whether the lists should be fused to a concept, and then asks what the name and descriptive words for that concept should be. Then, on the basis of the concept-label words, other possible candidate lists will be displayed and the process iterated until the expert is satisfied. At that point quantitative estimates of documents associated with each labeled concept can be made.

Enhancing the Method and Developing New Applications

The research reported in this paper is part of an ongoing, long-term effort. We are working to develop new techniques that will improve the existing technology. In the area of digital images, for example, so far we have clustered on the basis of either the spectral intensities associated with each pixel (as with the Landsat data) or the texture surrounding a pixel (as with the CT data). If we were able to combine both types of information, the cluster analysis might result in more subtle distinctions, such as trees in a suburban neighborhood versus trees in a forest. We are also experimenting with methods of generalizing the concepts extracted from an image of one area to images of other areas. Such a technique would allow the analyst to identify a concept, say, "deciduous forest," in one or two training images and then have any deciduous forest automatically identified in other images. Because every image produces a unique set of clusters, the success of this technique may hinge on our ability to map concepts to clusters identified by their relationship to other clusters, rather than by their specific centroid values.

We also hope to extend our methods to other data domains. Within the area of digital-image processing, we are exploring the analysis of x-ray images, and within the area of text processing, we are working with datasets pertaining to arms control and physics research. A logical new domain to tackle is that of one-dimensional signal analysis so that sound or sonar spectra can be clustered and mapped to concepts. A more challenging extension of our method would be to process data in which several different types of information—character, categorical, scalar—are recorded for each event in a dataset. For example, an intrusion detection program might analyze computer audit records containing the name of the user and the process (character), the duration of the process (scalar), and the error status of the process (categorical) for each completed process. A representation that combines these different types of information in a form that allows

clustering would, at a minimum, allow more efficient handling of typically huge audit records.

We view our concept-extraction technique as a bridge between imprecise human concepts and the world of physical, quantitative measurements. The method addresses a major weakness in the field of artificial intelligence: Although machine-learning algorithms enable computers to master real data, they fall short of being capable of what humans think of as intelligent behavior because the results of the learning process do not reflect human concepts. Rather than try to solve this problem directly, we have developed a tool that uses a partial statistical analysis to facilitate the mapping of concepts onto data. Until researchers develop a more fundamental solution that enables computers to discover human-relevant concepts on their own, concept extraction will allow us to make intelligent use of the massive amounts of data already being collected. ■

Further Reading

A. K. Jain and R. C. Dubes. 1988. *Algorithms for Clustering Data.* Prentice Hall.

W. Labov. 1973. The boundaries of words and their meanings. In *New Ways of Analyzing Variation in English.* Georgetown University Press.

P. M. Kelly and J. M. White. Preprocessing remotely-sensed data for efficient analysis and classification. In *SPIE Applications of Artificial Intelligence 1993: Knowledge-Based Systems in Aerospace and Industry.* 1993: 24–30.

Ching Y. Suen. 1979. *N*-gram statistics for natural language understanding and text processing. In *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 1: 164–172.

Roy E. Kimbrell. 1988. Searching for text? Send an *n*-gram! *Byte*. May, 297–312.

Patrick M. Kelly, James M. White, Judith G. Hochberg, Timothy R. Thomas, and Vance Faber

Vance Faber received his B.A. and M.A. degrees in mathematics in 1966 and 1969 and his Ph.D. in combinatorial group theory in 1971 from Washington University in St. Louis. After nine years as a professor at the University of Colorado at Denver, he accepted a position as a staff member in the Laboratory's Computer Research and Applications Group. Faber has been the leader of that group since 1992. He is extremely interested in novel applications of theoretical mathematics to problems that have societal impact.

Judith G. Hochberg received her B.A. in linguistics from Harvard University in 1982 and her M.S. and Ph.D. from Stanford University in 1985 and 1986. She came to the Laboratory in 1989 as a director-funded postdoctoral fellow, and since 1991 she has worked in the Computer Research and Applications Group. Before joining the Laboratory, Hochberg published research on sociolinguistics and child language learning with a particular interest in phonological rules. Hochberg's current research interests include various topics in computational linguistics as well as work in computer security and anomaly detection.

Patrick M. Kelly came to the Laboratory as a graduate research assistant in 1990, and in 1992 he joined the Computer Research and Applications Group as a technical staff member. He has participated in various projects, including Landsat data analysis, automated quantitative analysis of medical imagery, the development of clustering algorithms for massive digital datasets, and digital-image comparison and retrieval. His primary research interests include image processing and pattern recognition. Kelly earned his B.S. in computer engineering from the University of New Mexico in 1990 and his M.S. in electrical engineering from the University of New Mexico, Los Alamos, in 1992.

Timothy R. Thomas received his B.A. in psychology from the University of California, Berkeley, in 1964 and his M.A. and Ph.D. from Tulane University in 1968 and 1969. He came to the Laboratory in 1986 as an Association of Western University Fellow and in 1989 joined the Computer User Services Group as a technical staff member. Thomas's current research efforts include implementing a system of delivering archival images of scientific papers to distributed users and developing computerized methods of extracting information from large corpora of text documents. His past research focused on neural networks as tools for mapping from speech to tongue movement. He has published research on perception, neural control of behavior, and hormonal physiology. Thomas has recently returned from Borneo where he assisted his daughter in observing orangutans for a Harvard University rain-forest-ecology project. He is currently preparing a 12-million-year-old rhinoceros skull that he found near Española, New Mexico, for display at Northern New Mexico Community College.

James M. White is section leader of the Computer Research and Applications Group and is responsible for designing and developing software applications in digital-processing techniques and applying these techniques to a wide range of scientific fields. He also initiates, executes, and manages new projects pertaining to image processing. White received his B.S. and M.S. in civil engineering from the University of Maryland in 1979 and 1982. Before joining the Laboratory in 1985, he was a Faculty Research Associate and Research Engineer at the University of Maryland and served as a Digital Imagery Scientist/Acting Chief of the Interactive Digital Image Manipulation Branch of the Central Intelligence Agency. White is a consultant in image and signal processing to government agencies, private firms, and universities. He was a member of the NASA Committee on Image Processing of the Shuttle Challenger Disaster in 1986. In 1988 White was issued a patent for a Monte Carlo vector quantizer.