JOHN Y. C. WANG
U. S. Army Engineer Topographic Laboratories
Fort Belvoir, VA 22060

Feature Extraction by Interactive Image Processing

The computer aids the operator in detecting and delineating cultural features on digitized imagery.

ABSTRACT: *In aerial photography, images of roads and other man-made structures are small compared to those of natural features. Their edges are usually indistinct from the background and often go undetected by automatic systems. A major problem encountered by automatic detection systems is the delineation of boundaries around areas containing features of interest. The computer is often unable to accomplish the detection process alone. Therefore, it is reasonable to combine the superior pattern recognition abilities of a human being with the computational power of a digital computer to form an interactive system. Such an interactive feature extraction system is discussed. The objective is to generate the photo coordinates for feature boundaries from aerial photographs. The hardware consists of a CDC-6400 sequential computer, a Goodyear STARAN associative array processor, a DEC PDP 11/50 minicomputer, and an image display, as well as digitizing and recording devices. The feature extraction program for the interactive system consists of a series of routines which were designed, written, and installed in the system by the author. The routines were written in FORTRAN and/or the STARAN assembly language according to the intended application in order to use the computers efficiently.*

INTRODUCTION

An IMPORTANT SUBJECT before scientists, engineers, and cartographers at the present time is the rapid, accurate, and economical processing of imagery that depicts the surface of the earth or other planets remotely sensed from space vehicles. In mapping we are interested in planimetric features such as roads, built-up areas, drainages, and natural boundaries; however, tracing planimetric features from aerial photographs rapidly, accurately, and economically is not a simple task. Presently, automated cartographic instruments can produce a variety of maps, but cannot compile line maps completely automatically. Cartographic details still must be extracted manually by human operators. The extraction of features is the bottleneck in the total automation of the mapping processes.

Image processing methods may convert the original picture into simpler form, e.g., a line drawing, by some sequence of steps. Each step should preserve the important properties of the original, while reducing the computational requirements imposed on subsequent steps. But the reduced picture generally contains less information than the original, and there is no guarantee that the information lost is irrelevant. A number of methods to determine planimetric features automatically from digital data may serve for some types of imagery or for some special purpose[3,4], however, there is no single standard approach to the extraction of features from remotely sensed data. In practice, it is not always wise to attempt to map objects into clear-cut, two-value overlays. Very often objects are defined only in a "fuzzy" sense and no definite or distinct edge can be drawn by automatic methods.

PHOTOGRAMMETRIC ENGINEERING AND REMOTE SENSING,
Vol. 43, No. 12, December 1977, pp. 1495-1501.

In aerial photography, roads and buildings are very small compared to fields and forests, and their edges usually are not distinct from the background. In small-scale photography, roads are usually imaged as a white line with a small surface area. These may often go undetected by automatic systems while features of less interest to map makers, such as fields, are easily detected. Even with a good edge extraction method, there remains the problem of deciding which edges should be extracted to give the maximum amount of information in the final map. Feature extraction from digital imagery is, in general, a complicated problem, so that the computer is often unable to accomplish the task alone. Therefore, it appears reasonable to combine the superior pattern recognition abilities of the human being with the computational power of the digital computer to form an interactive system. In this paper we are concerned with the extraction of features by the interactive digital image system in order to generate photo coordinates automatically. If we have the photo coordinates of features from stereo photography, we can use analytical photogrammetric methods to generate ground coordinates and to compile line maps.

Interactive Digital Image Processing System

An interactive digital system offers flexibility. Once the basic hardware and software systems are developed, only minimal software effort is required to utilize the system for new applications. The interactive digital image processing system which was utilized to conduct this research was developed at the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia. The hardware consists of a CDC-6400 sequential computer, a Goodyear STARAN associative array processor, a DEC PDP 11/50 minicomputer, standard peripherals, and associated image display, digitizing, and recording devices. The CDC 6400, STARAN, and PDP11 computers have their own system software package to handle all operations for their system. There are interface software packages to allow all computers in the system to communicate with each other.

Images to be processed on the interactive digital image processing system are input to the system from magnetic tape or digitized transparencies made on the DICOMED D56. Once the images are in the system, they are stored on CDC's 844 disks. The interactive capability is achieved by users working through the Tektronix 4014-1 alphanumeric console and the COMTAL image display unit. The computations are done in the CDC-6400 computer and STARAN parallel processor that allows the programmer to take advantage of parallelism in the arithmetic structure of a problem. The final results can be put on hard copies, printed on transparencies on-line by the DICOMED D47, or off-line by copying to magnetic tape for recording on other hard copy devices.

Basic Algorithms

A typical picture usually consists primarily of features having homogeneous densities and certain properties, such as size, shape, direction, orientation, and adjacency which are most important for map makers. If we can accurately determine the boundaries (edges) of the objects, we may be able to measure these essential properties. We assume that the boundary around an object is a closed loop except as it reaches the limits of the data. The following algorithms to extract the features by using the interactive digital image systems have been developed.

Edge Detection

Initially, the user has to provide a starting point and a threshold value for detecting edges after selecting a feature for manipulation. The basic approach that we have designed consists of two steps. The first is finding an initial point. The computer compares the gray level of one pixel at a time of a small array, whose upper-left corner is the given starting point, against the input threshold value. If the threshold gray level lies between the gray levels of two adjacent pixels, an edge exists between the two pixels. If the two adjacent points are on the same line, the orientation of the edge is vertical across the length of the pixel. If the two adjacent points are on the different lines, the orientation of the edge is horizontal.

The second step is detecting a new edge point by comparing the gray levels of three nearest neighbors of the initial point against the threshold value. It is done according to the orientation (horizontal and vertical) of the edge. Suppose (see Figure 1) a horizontal edge point is at location ⊗ between elements (i, j) and $(i+1, j)$ and assume the gray level of element $(i, j) \geq T$ (the threshold value) and the gray level of element $(i+1, j) \leq T$. If the edge has an extension, there are three cases; the new edge point may be (a) between elements (i, j) and $(i, j+1)$, (b) between elements $(i, j+1)$ and $(i+1, j+1)$, or (c) between elements $(i+1, j+1)$ and $(i+1, j)$. Only two neighbor elements

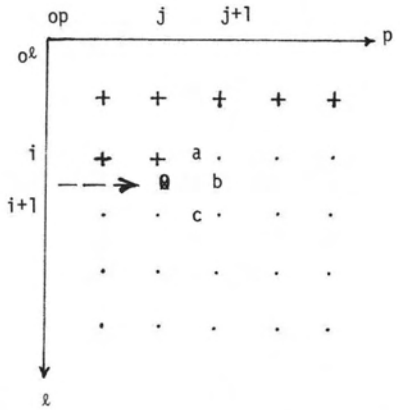

FIG. 1. Detecting a new edge point from a known horizontal edge.

need be checked to determine the location of the edge. If the gray level of element $(i, j+1) \leq T$, it is case a. If it is not case a, and the gray level element $(i+1, j+1) \leq T$, it is case b; if the gray level of element $(i+1, j+1) > T$, it must be case c.

If the previous edge is vertical and the last edge point is at location ⊗ (see Figure 2) and assuming the gray level of element $(i, j) \geq T$ and the gray level of element $(i, j+1) \leq T$. There are three cases. The edge may be (a) between the elements (i, j) and $(i+1, j)$, if the gray level of element $(i+1, j) \leq T$; (b) between the element $(i+1, j)$ and $(i+1, j+1)$, if it is not case a and the gray level of element $(i+1,j+1) \leq T$; and (c) between the element $(i, j+1)$ and $(i+1, j+1)$, if it is neither case a nor case b.

LINE DETECTION

A line is a long and narrow region which has a certain width, homogeneous gray levels in the region, and a gray level distinctly different from its background. It is difficult to detect a line by the edge detecting method; but if we can determine the width and the central points of the line, we may be able to measure the essential properties. Initially the user must provide a starting point, the critical value (IC) of the gray level difference between adjacent elements in the line region, after he has selected a line feature for manipulation. There are two steps for detecting. The first step is searching all parameters which are required in the second steps, such as the width, direction, orientation, and the gray level of the line. The second step is locating the central points of the line by comparing the difference between the gray level of elements and those of the background against the critical value (IG) and also comparing the difference of gray level between adjacent elements along the line against the critical value (IC) according to the orientation of parameter of the line. If the difference of the average gray level of the cross section of the line and the average gray level of background is greater than the critical value (IG) and the difference of the average gray level of two adjacent cross sections of the line is smaller than the critical value IC, then the new points are the line points and the central point of the cross section is recorded.

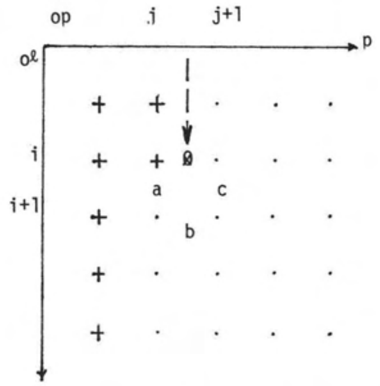

FIG. 2. Detecting a new edge point from a known vertical edge.

PROCEDURES

A stereo pair of aerial photographs #98 and #99 over the Arizona Test Area were used. The scale of photography is about 1:47,100. The Model 1050A Microdensitometer system was used to measure and scan the images for this research. A point in the upper-left corner of the photo was chosen as the origin of the photo coordinates system. For both photos, all fiducial marks, the *reseau* intersections of interest, and the upper-left corner of the mapping area which would be digitized were measured with the scanner in the manual mode, and the photo coordinates were recorded. An aperature size of 34.5 micrometers and the 5× magnification for the eyepiece and the objective lens were chosen. Then scan line and pixel spacings were selected to be 30 micrometers each. A square array (2048 × 2048 points) on each photo was digitized, and the data were recorded on magnetic tapes. After inputting a portion of the digital data from the photography into the system and displaying it on the screen, the user can examine the scene and decide which feature will be extracted first. Then the subset of data which contained the feature will be moved into the computer and displayed on the screen at a larger scale.

EDGE DETECTION

If an edge is to be detected, the user should input a starting point and a threshold value. The starting point can be entered in two ways: one is the coordinates of the point, the other is the position of the cursor on the screen. There are three ways to obtain the threshold value: (1) Users will provide the threshold value which must be input to the system; (2) the program will provide the threshold value by using the mean gray level of the data; or (3) by plotting the histogram of the gray levels of the data and using the bottom value of the valley of the histogram.

The first step is to search for an initial edge point in a small array in which the upperleft corner is the given starting point. The computer compares the gray level of one pixel at a time against the threshold value. If the threshold gray level lies between the gray levels of two adjacent pixels, an edge exists between the two pixels. After finding an edge point, the program will stop the search and record the local coordinates of the initial edge point for the edge and the orientation of the edge. The local coordinates of the initial edge point (refer again to Figures 1 and 2) are $p = op + j + 0.5$ and $l = ol + i$ for the vertical edge and $p = op + j$ and $l = ol + i + 0.5$ for the horizontal edge, respectively.

The second step is to locate the edge points according to the orientation of the edge. If there is a horizontal edge, the local photo coordinates of the edge points are calculated as follows (see Figure 1):

(a) If it is case a, the location of the new edge point is between elements (i, j) and $(i, j+1)$; then the local coordinates of the point are

$$p = op + j + 0.5 * \text{IFG}$$

and (1)

$$l = ol + i.$$

(b) If it is case b, the location of the new edge point is between element $(i, j+1)$ and $(i+1, j+1)$; then the local coordinates of the point are

$$p = op + j + 1$$

and (2)

$$l = ol + i + 0.5.$$

(c) If it is case c, the location of the new point is between element $(i+1, j)$ and $(i+1, j+1)$; then the local coordinates of the point are

$$p = op + j + 0.5 * \text{IFG}$$

and (3)

$$l = ol + i + 1.$$

op and ol are the coordinates of the upper-left corner point of the data matrix. IFG is the indicator of edge direction. If IFG=1, the edge is from left to right. If IFG=−1, the edge is from right to left. The unit of the p and l is the pixel spacing of the digitized data.

For detecting an edge point from a known vertical edge and a threshold value, the principal detecting method and procedure are the same as the known horizontal edge. The difference is the orientation of the known edge; therefore, the comparison is done counterclockwise and the ways for generating the local coordinates are different.

The local coordinates of edge points are calculated as follows (see Figure 2):

(a) If it is case a, the location of the new edge point is between element (i, j) and $(i+1, j)$; then the local coordinates of the point are

$$p = op + j$$

and (4)

$$l = ol + i + 0.5 * \text{IFG}.$$

(b) If it is case b, the location of the new edge point is between element $(i+1, j)$ and $(i+1, j+1)$; then the local coordinates of the point are

$$p = op + j + 0.5$$

and (5)

$$l = op + i + 1.$$

(c) If it is case c, the location of the edge point is between the elements $(i+1, j+1)$ and $(i, j+1)$; then the local coordinates of the point are

$$p = op + j + 1$$

and (6)

$$l = ol + i + 0.5 * \text{IFG}.$$

op and *ol* are the coordinates of the upper-left corner point of the data matrix. IFG is the indicator of edge direction; if IFG = 1, the edge is from top to bottom; if IFG = −1, the edge is from bottom to top. The unit of the *p* and *l* is the pixel spacing of the digitized data.

LINE DETECTION

If a line is to be detected, the user should provide a starting point and two critical values (IG and IC). The program will first search the area around the starting point to generate the property parameters such as width, direction, orientation, and the initial central point of the line by counting the elements whose difference of gray levels is smaller than the critical value IC. If the horizontal length of the region is longer than the vertical length, the line is orientated horizontally and the vertical length is the width of the line; the middle point of the cross section is the central point of the line. If the horizontal length is equal to the vertical length, it means the line is a 45-degree tilted line. The parameters for the tilted line are obtained in the same manner as for the horizontal or vertical line. After the program has all parameters, it detects the line according to its orientation. If it is a vertical line, the average gray level of the horizontal elements of the line (the cross section of the line) is compared to the average of two background neighbors. If the difference of the average gray level of the line and average gray level of the background is greater than the critical value IG and the difference of the average gray level of the line and average gray level of the background is greater than the critical value IG and the difference of the average gray level of two adjacent cross section of line is smaller than the critical value IC, then the new point is the line point (see Figure 3). The mathematical expressions are as follows:

$$\left|\frac{1}{IW}\sum_{j=1}^{IW} g(i, j+k) - \frac{1}{2}\left[g(i, k) + g(i, k+IW)\right]\right| \geq \text{IG} \quad (8)$$

and

$$\left|\frac{1}{IW}\sum_{j=1}^{IW} g(i, j+k) - \sum_{j=1}^{IW} g(i+1, j+k)\right| \leq \text{IC} \quad (9)$$

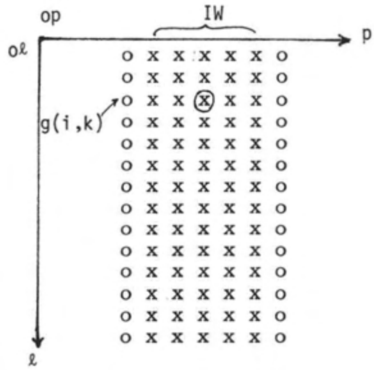

FIG. 3. The vertical line.

IW is the width of the line and $g(i, k)$ is the left nearest background neighbor of the line section.

Equations 8 and 9 must be satisfied simultaneously. The cross section at i is a part of the line and the central point is at $g(i, k + (IW/2))$. If some cross section fails the Equation 8 test and passes the Equation 9 test, the program will not register the cross section as a part of the line and will process the next cross section until some section fails both tests, then the program terminates the line.

The program will generate the local coordinates for the central points of a line by using Equation 10.

$$p = op + k + IW/2$$

and (10)

$$l = ol + i$$

where *op* and *ol* are the local coordinates of the upper-left corner of data matrix, and *IW* is the width of line.

The principles of and procedures for detecting other orienting lines are the same as those for the vertical line. The only difference is in the equations for testing and calculating the local coordinates; therefore, that the details are omitted.

All edge or line points are displayed on the screen by an overlay. If no additional new points for the edge or line can be located, or the edge or line has reached the limit of the data, the process is paused. Operator assistance is needed to give a new starting point or perform other operations. If a boundary of an edge or line is terminated, the local coordinates can be output. However, because the edges or lines are usually smooth curves, the two-dimensional raw data (local coordinates) might be smoothed as a straight line or a second-order curve according to their shape by a least square adjustment program. After adjustment, the results are displayed on the screen; then the user may decide how many points will be used and/or stored as the local coordinates for representing the feature of the photograph.

The local coordinates of corresponding feature points can be obtained by using a digital image match technique. An independent array (11 × 11) whose central point is the known feature point will move around on the corresponding image to find a match. The correlation values are computed according to the following equation for matching:

$$CC_{AT}(m, n) = \sum_{ij}\sum (AD*TD)/(\mathrm{VAR}[A]*\mathrm{VAR}[T])^{1/2} \quad (11)$$

where A is the dependent array, T is the independent array,

$$AD = [A(i, j) - AM],$$
$$TD = [T(i - m, j - n) - TM],$$
$$\mathrm{Var}\,[A] = \sum_{ij}\sum (AD)^2,$$

and

$$\mathrm{Var}\,[T] = \sum_{ij}\sum (TD)^2.$$

Var $[A]$ and Var $[T]$ are variances of A and T, and AM and TM are the mean of A and T, respectively.

The routine then finds the maximum value among the correlation values and its eight nearest neighbors, and it fits a second-order surface through the nine points by a least square adjustment routine. The adjusted peak of the surface is the matched point (see Figure 4).

The coordinates of the matching point are:

$$p' = p + \Delta p + \epsilon p$$

and (12)

$$l' = l + \Delta l + \epsilon l.$$

where p and l are estimated coordinates, Δp and Δl are shifts between the estimated p and l coordinates of the largest computed correlation location, and ϵp and ϵl are the corrections of coordinates from the least squares adjustment. p' and l' are the final local coordinates for the corresponding feature points.

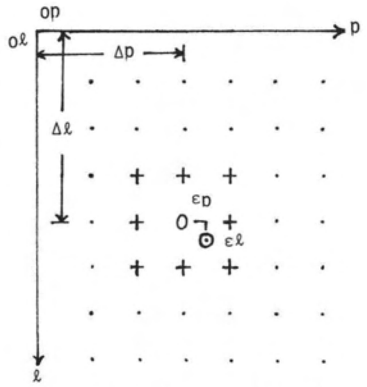

FIG. 4. Two dimension correlation function and the match point.

PHOTO COORDINATES

The photo coordinates of features are generated:

$$x = ULX + p * s - ox$$

and (13)

$$y = -(VLY + l * s - oy).$$

Where ULX and VLY are the plate coordinates of the upper-left corner of the digitized area; p and l are the local coordinates of feature; s is the scale factor (the pixel spacing); and ox and oy are the plate coordinates of the principal point. The photo coordinates of the features accompanying descriptive title or heading on each file and a series of codes for each line segment are a part of the output. The codes will indicate the type and kind of features. One code will indicate whether the feature is an edge or a line. If it is a line, the width of the line should be recorded, since the line detecting routine only gives the coordinates of the central line. If it is a closed boundary, a code indicating the texture within the region is needed. Another code will indicate whether the data represent a straight line or a curve. If it is a curve, the mathematical model is used for the least square adjustment routine. From the codes, the map compiler can determine precisely the characteristics of the feature without referring to the photograph.

CONCLUSION

The tracing of features manually in great detail is time-consuming and requires skilled operators. Since these tasks are routine, repetitive operations that require great precision, it would be desirable to automate the entire process. However, many of these tasks require perceptual abilities and are expected to continue to be beyond the capabilities of a fully automatic system for the near future.

An alternative approach is to combine abilities of man and machine—an interactive system. This approach has several advantages.

- Decisions can be made and adapted to the problem as frequently as the operator likes. When errors are committed, he may interrupt the process to make corrections and/or to improve the results until he is satisfied.
- Techniques can be selected depending on the context. If a routine procedure does not work in a particular situation, the operator may try various techniques to process the data. This flexibility gives researchers a very useful tool for experimenting with various techniques and procedures for evaluating data for exploratory development.
- Only the data of immediate interest to the user will be handled. As a result, less space will be required and computer time will be reduced. The computations which are inherently parallel in nature were performed on the STARAN parallel processor to speed up the processing.

This paper demonstrates the capability for extraction of cartographic detail from digital aerial photographic data. It appears to offer a substantial improvement in speed over the manual method. Also, its interactive capability between operator and computer offers a great potential for applications in other fields.

References

1. Crombie, Michael A., *Semiautomatic Pass Point Determination Using Digital Techniques*, ETL-0051, U.S. Army Engineer Topographic Lab., Ft. Belvoir, VA 22060, December 1975.
2. Gambino, L. A. and M. A. Crombie, "Digital Mapping and Digital Image Processing," *Photogrammetric Engineering*, November 1974.
3. Ketting, R. L. and Lundgrebe, *Automatic Boundary and Sample Classification of Remotely Sensed Multispectra Data*, LARS Information Note 041773, LARS Purdue University, April 1973.
4. VandeBrug, G. J., "Line Detection in Satellite Image," Machine Processing of Remotely Sensed Data Symposium, *Proceedings*, LARS Purdue University, June 1975.
