N91-15616

Refinement of Ground Reference Data With Segmented Image Data

Jon W. Robinson
Image Analysis Section
ST Systems Corporation
4400 Forbes Boulevard
Lanham, MD 20706
Phone: (301) 794-5200
E-Mail: mscc@dftnic.gsfc.nasa.gov

James C. Tilton
Mail Code 936
Information Systems Development Facility
NASA Goddard Space Flight Center
Greenbelt, MD 20771
Phone: (301) 286-9510
E-Mail: tilton@chrpisis.gsfc.nasa.gov

ABSTRACT

There are a variety of ways for determining ground reference data for satellite remote sensing data. One of the ways is to photo-interpret low altitude aerial photographs and then digitize the cover types on a digitizing tablet. These digitized cover types are then registered to 7.5 minute U.S.G.S. maps that have themselves been digitized. The resulting ground reference data can then be registered to the satellite image, or, alternatively, the satellite image can be registered to the ground reference data. Unfortunately, there are many opportunities for error when using a digitizing tablet and the resolution of the edges for the ground reference data depends on the spacing of the points selected on the digitizing tablet. One of the consequences of this is that when overlaid on the image, errors and missed detail in the ground reference data become evident. This paper discusses an approach for correcting these errors and adding detail to the ground reference data through the use of a highly interactive, visually oriented process. This process involves the use of overlaid visual displays of the satellite image data, the ground reference data, and a segmentation of the satellite image data.

Several prototype programs have been implemented on the VAX computer system and the IVAS image display system to examine various methodologies for improving ground reference data. These programs provide a means of taking a segmented image and using the edges from the reference data to mask out those segment edges that are beyond a certain distance from the reference data edges. Then using the reference data edges as a guide, those segment edges that remain that are judged not to be image versions of the reference edges are manually marked and removed. We describe the prototype programs that were developed and the algorithmic refinements that facilitate execution of this task. Finally, we point out areas for future research.

INTRODUCTION

There are many ways to use multiple data sources in remote sensing. In this paper, we discuss a method for using image data to improve ground reference data sets. Reference data sets are sometimes referred to as "ground truth"; however, since the methods of creating reference data sets provide many opportunities for error, we will use the term reference data set. This terminology allows for the possibility of using multiple data sources for determining not only the contents of the reference data set but also for refining it.

This work was supported by the Office of Aeronautics, Exploration and Technology, NASA Headquarters, Washington, DC.

We have obtained satellite image data from a number of investigators along with reference data sets that they created. By using the image data to create a segmented image, edges consistent with the spectral variation in the image are created. Visual inspection of the reference data edge map overlaid on the raw image or the segmented image reveals many discrepancies. These may be errors, or simply inappropriate attention to detail by the person generating the reference data (particularly if they were digitizing data from a digitizing tablet). An example exhibiting displacement errors and too coarse of a digitization scale is given in Figure 1.

While it would be desirable to have an automatic method of using the segmented image data to revise the reference data set, we chose, as a first approximation, to develop prototype methods that allow interactive selection and deletion of the segmented image edges that appeared to be too far from those edges generated from the reference data set. The end result is to leave a set of edges that are considered to be the true edges derived from the segmented image data, based on their proximity to the original reference data set edges.

METHODS

A series of programs were developed using IDL (Interactive Data Language) that allowed the user to use either an IIS system 575 terminal or an IIS IVAS terminal attached to a VAX cluster for interactive editing of the edge map generated by an image segmentation algorithm (Tilton, 1989) which runs on the MPP (Massively Parallel Processor) at Goddard Space Flight Center.

The image segmentation algorithm is an iterative process. At each iteration, those regions that are most similar by a particular criteria (*e. g.*, minimum change in image entropy, or minimum rise in image mean squared error) are merged. As the number of iterations increase, the number of image segments decreases. A program called "edge.movie" that runs on the IIS System 575 was developed that allows the user to view various iteration steps and compare these with the original reference data set edges and bands from the image data.

This program allows the user to interactively pick an iteration and then immediately compare it with these other sources of information. It is best to select an iteration with more segments and edges than necessary to fit the reference data so that a crucial segment edge that may match a particular reference edge will not be lost. On the other hand, choosing too low an iteration and thus an image with too many segments and edges increases the work load of the analyst significantly.

The next step is extract the edges of the selected iteration from the combined edge file which contains the edges from all iterations coded by iteration number. This leaves an image of all of the edges present at iteration n with the pixel value of the edge indicative of its age.

The original reference data is a raster format image divided into regions. An IDL function, GREFEDGE.PRO was written to extract the edges from this image.

This segmented edge file is then masked with the edge file from the reference data to eliminate all edges that are beyond a specified distance from the reference set edges. This distance is selected by the analyst and depends on the image set being analyzed. Figure 2 shows an example of this process after this masking is complete.

Because the main algorithm for this procedure can only process images that are 128 x 128, sections of this size or smaller must be extracted from larger images for processing and then be recombined to produce a final product. In order for there to be continuity between sections it is necessary to have a certain amount of overlap between them. The degree of overlap required is being investigated in ongoing studies.

Two programs, CHGVAL5.pro and CHGVAL4.pro (for the IIS System 575 and IIS IVAS respectively) were written in IDL to provide a prototype interactive environment (listings in Appendix A). The input files (image, segment edge, original reference edge) for either of these programs can either be input one at a time as the program requests or the input file names can be read from an input file.

The CHGVAL5.pro program for the IIS System 575 image display system uses a track ball to move the cursor on the image display and the track ball function buttons to either delete an edge pixel, replace and edge pixel or go on to the next step in the program.

The CHGVAL4.pro program for the IIS IVAS system functions the same way except that it allows the display to be zoomed and roamed when looking for edges to delete. In addition, the IIS IVAS system uses a three button mouse instead of a multi-button track ball.

The CHGVAL#.pro programs start by asking whether the analyst wishes to enter the names of the image files individually or whether an input file containing all of the other input items is to be used. Since the program is usually invoked many times to complete the processing of a single section, the use of the input file saves the analyst the trouble of remembering and typing in all of the other file names each time the program is used on a section.

The edges from the original reference image are loaded into the graphics plane of the display device. Two bands of the original image data are loaded into the red and green display memories and the edges from the masked segmented edge image are loaded into the blue display memory.

In the CHGVAL5.pro program for the IIS system 575, the analyst uses the track ball to move the cursor over parts of the blue edge image that they wish to remove and click on the appropriate track ball button, producing "nicks" in the selected edge. For the CHGVAL4.pro program for the IIS IVAS system, the user can zoom and roam the image with the mouse first so it is easier to see what and where one is deleting edge segments. The terminal monitor prompts the user with the current functions of the track ball buttons or the mouse buttons. An example of this process after nicking several edges is given in Figure 3.

After having nicked a number of edges for removal, the analyst can exit this part of the program and the nicked edge image is written out to a file for processing by a batch program that runs on the MPP. After writing out the nicked edge image, the CHGVAL#.pro program submits the batch program to the MPP and waits for it to finish. The MPP program writes out the new edge image with the nicked edges completely removed. The CHGVAL#.pro program then reads in the new edge image and writes it out to the red display memory so the analyst can review the results of his work.

When the new edge image is read into the red memory bank, those parts of the original edge image that were deleted show up as blue and those parts that were not deleted show up as magenta (red "new edge image" + blue "blue old edge image"). The analyst is then given the choice of undoing some of his deletions or continuing with further nicking or quitting and saving the result. An example of the final result from CHGVAL#.PRO for a single 128 x 128 section of data is given in Figure 4.

There are two types of edge connectivity that can be used, 8 connected and 4 connected. With 8 connected edges, unexpected deletions can chain through edge intersections. Thus it is particularly useful to have a number of ways of back tracking.

Our prototype system provides several alternative ways to recover edges. The simplest method is that while nicking lines, the user can undo a nick by placing the cursor over the nicked pixels and pushing the appropriate key on the mouse or the track ball. Alternately, the user can run a test, deleting the nicked edges and determine whether he wants to undo some of the nicks. If the user chooses to undo some of the nicks he can either fill in particular nicks using the cursor control device or he can pop deleted pixels off of the stack into their former locations. The latter method is most effective if the most questionable deletes are saved until near the end of a nicking cycle.

These methods of recovering erroneously deleted edges only work within a single nick and try cycle. If one either continues with a new nick and try cycle or saves the last result and exits the CHGVAL#.PRO program, then another method must be used to recover lost edges. In order to repair or recover edges after leaving the CHGVAL#.PRO a program that would allow pasting edge pixels from an earlier edge image into the current edge image was developed. This program puts one band of the image into the green display memory, the edge image to be fixed into the blue display memory and the master or original edge image into the red display memory. The user then uses the cursor control device to put the cursor over the master edges that he wishes to copy to the edge image to be fixed and presses the appropriate button. This puts a copy of the master pixel overlayed by the cursor into the edge image to be fixed. In this manner each pixel of an edge segment in the master image can be copied into the edge image to be fixed. An appropriate change in color takes place for each pixel that is moved into the edge image that is being repaired so the user can tell which pixels are present in both images.

The above description is summarized in Figures 5 and 6. Figure 5 illustrates the overall data flow from the original image and ground reference data to the revised reference edges (producing a revised ground reference data file). Figure 6 is a flow chart describing the CHGVAL#.PRO programs.

After several overlapping 128x128 edge images have been edited, it is necessary to rejoin them into a single image, to delete nonterminating edges in the overlap region and to join edges from overlapping segments. Another IDL procedure, COMBCHG4.PRO that runs on the IIS IVAS display is being developed to accomplish this task. Since its functions are so similar to those of CHGVAL4.PRO it is being designed to perform these functions also.

Two test images are being used. The first is a 468 x 368 pixel Thematic Mapper (TM) image of the Ridgely Quadrangle on the eastern shore of Maryland. It contains few classes and most of them are large in area. The main features of the scene are fields and wooded streams. There are small areas of water, ponds, and single urban area. These areas were digitized from a 7.5 United States Geological Survey quadrangle map that had had its reference data boundaries drawn on a plastic overlay from 1977 aerial photographs using a digitizing tablet (Gervin, et al., 1985). In the original study, the ability of AVHRR and MSS data to distinguish Level 1 land cover classes was examined. This involved registering the MSS data to the Ridgely Quadrangle and resampling it to 60 meter pixels. This lead to the ground reference data being rasterized to 60 meter pixels. Since the pixel size of the original reference data set was 60 meters and TM data is 30 meters, the disparity between the TM boundaries and the reference data boundaries, and lack of detail in the reference data are understandable. Examples shown in Figures 1 through 4 are from a 128 x 128 section in the upper left corner of this data set.

The second image is a portion of the Washington D.C. metropolitan area. This area was broken down into more classes than the Ridgely quadrangle and the classes are smaller in size. Tests with this data set were not yet completed as of this writing.

RESULTS & DISCUSSION

In practice, the prototypes worked well, allowing the edge maps to be trimmed by nicking those edges to be deleted. Problems were primarily ones of speed in loading the original reference data into the graphics plane.

The use of 8 connected edges lead to unexpected chains of deletions. Because this was so severe, the processing procedures were started over using 4 connected edges.

Joining the segments with overlapping boundaries was not a serious problem with the Ridgely quadrangle. The large areas and concomitant small number of edges contributed to the each in joining the segments together.

Looking again at Figure 4, note that refined ground reference boundaries (obtained by selecting boundaries from the image segmentation) follow very closely visible boundaries in the image data. In particular, note the very coarse, misregistered boundary from the original ground reference file the top middle of the image. The refined ground reference boundary is perfectly registered, and follows the actual variations in the boundary very closely.

We plan to use this and other data sets in comparative studies of various algorithms designed to extract spatial information from imagery. We expect that the refined ground reference data will help to more accurately evaluate the behavior of these algorithms than would the often too coarse and misregistered original ground reference data.

REFERENCES

Gervin, J. C., A. G. Kerber, R. G. Witt, Y. C. Lu and R. Sekhon, 1985, "Comparison of Level I Land Cover Classification Accuracy for MSS and AVHRR Data," *International Journal of Remote Sensing*, 6(1), pp. 47-57

Tilton, J. C., "Image Segmentation by Iterative Parallel Region Growing and Splitting," *Proceedings of the 1989 International Geoscience and Remote Sensing Symposium*, Vancouver, BC, Canada, July 10-14, 1989, pp. 2420-2423.

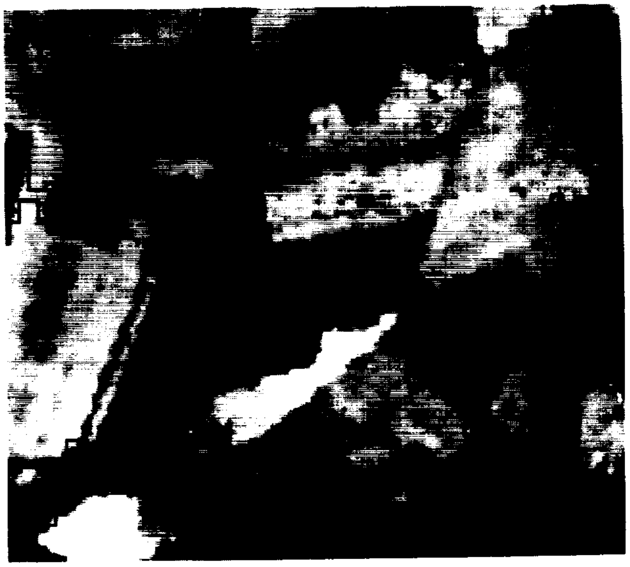

Figure 1. Edges from a ground reference file (black) overlaid upon the corresponding Landsat Thematic Mapper (TM) image. Note the displacement errors and coarse digitization scale compared to the Landsat data.

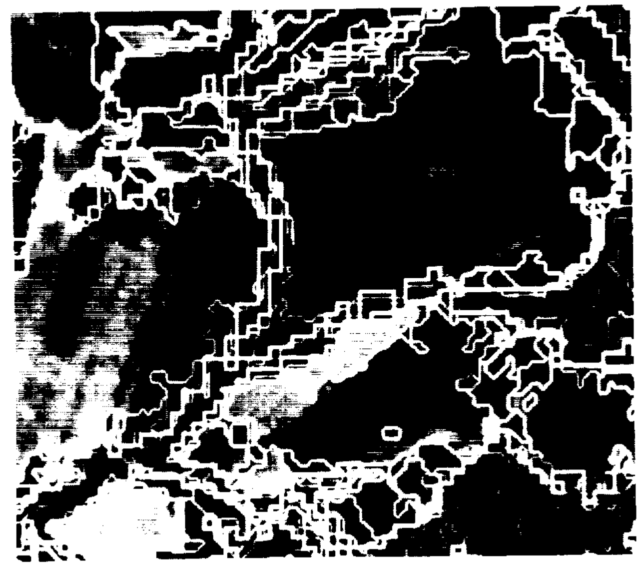

Figure 2. Edges from a ground reference file (white) and edges from an image segmentation (gray) overlaid upon the corresponding Landsat TM image. Image segmentation edges further than 6 pixels from a ground reference edge have been masked out.

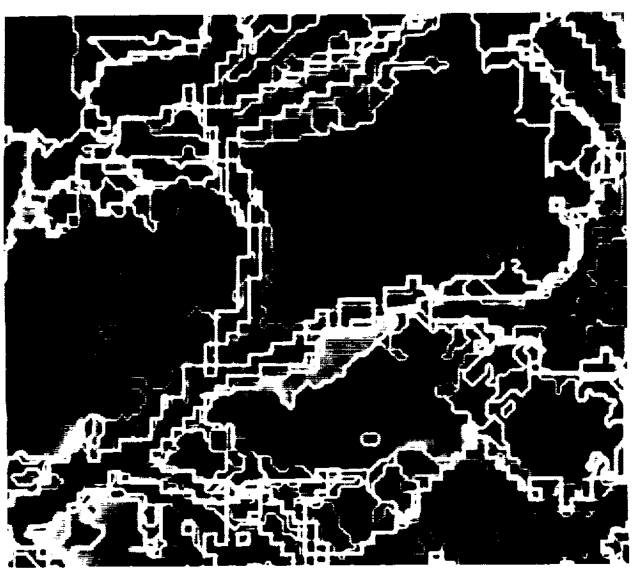

Figure 3. Edges from a ground reference file (white) and edges from an image segmentation (gray) overlaid upon the corresponding Landsat TM image. Image segmentation edges that are shown to be "nicked" in this image will be deleted by the next processing step (connected components labeling).

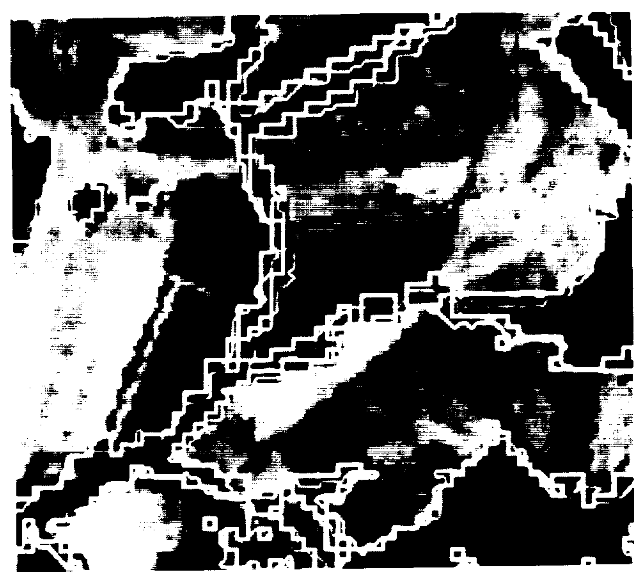

Figure 4. Edges from a ground reference file (white) and the final selection of corresponding edges from an image segmentation (gray) overlaid upon the Landsat TM image. This final selection of edges can now be used to generate a label map that can be used as a substitute for the original ground reference file.

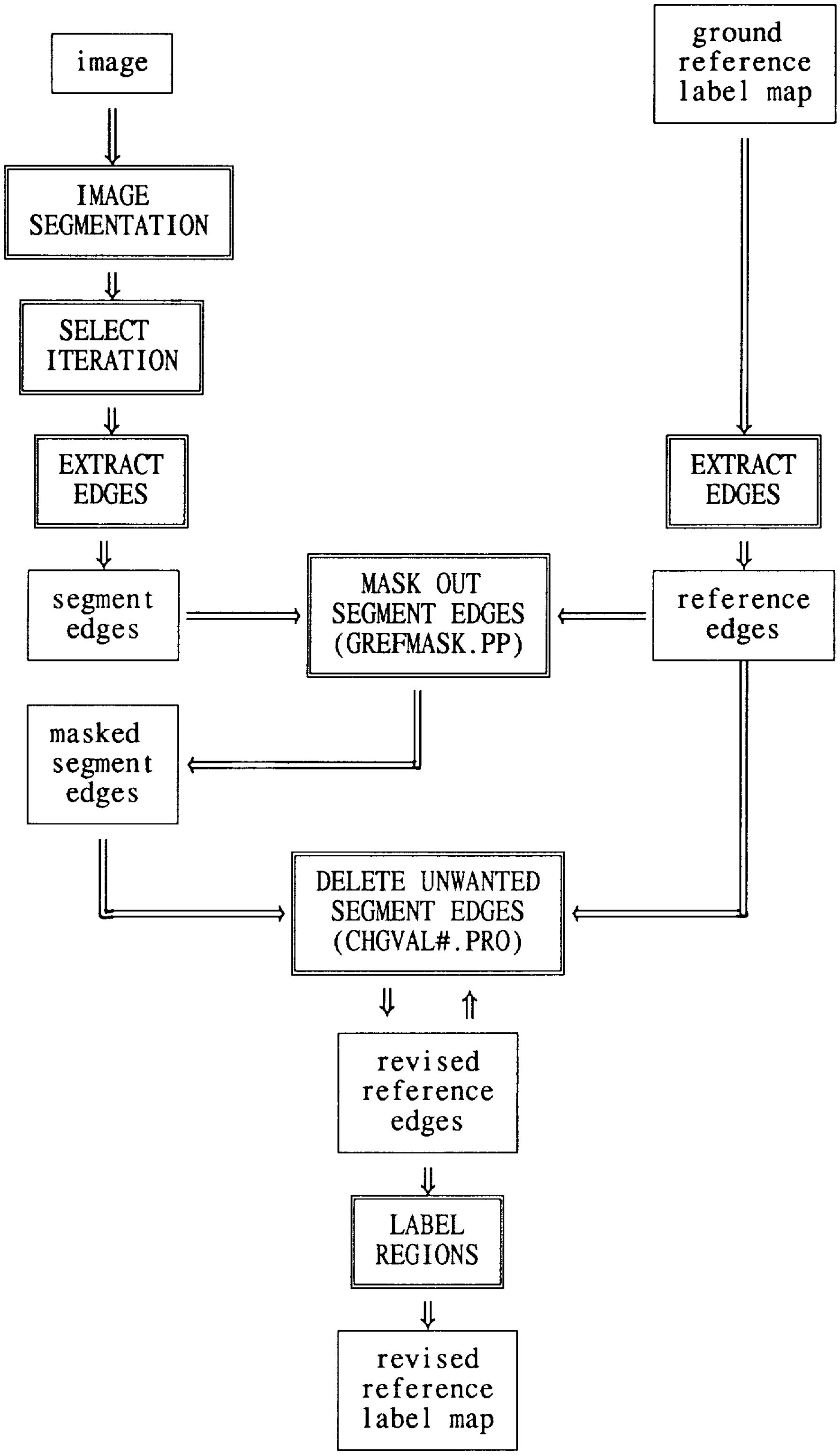

Figure 5. Overall data flow for producing a revised ground reference label map based upon an image segmentation.

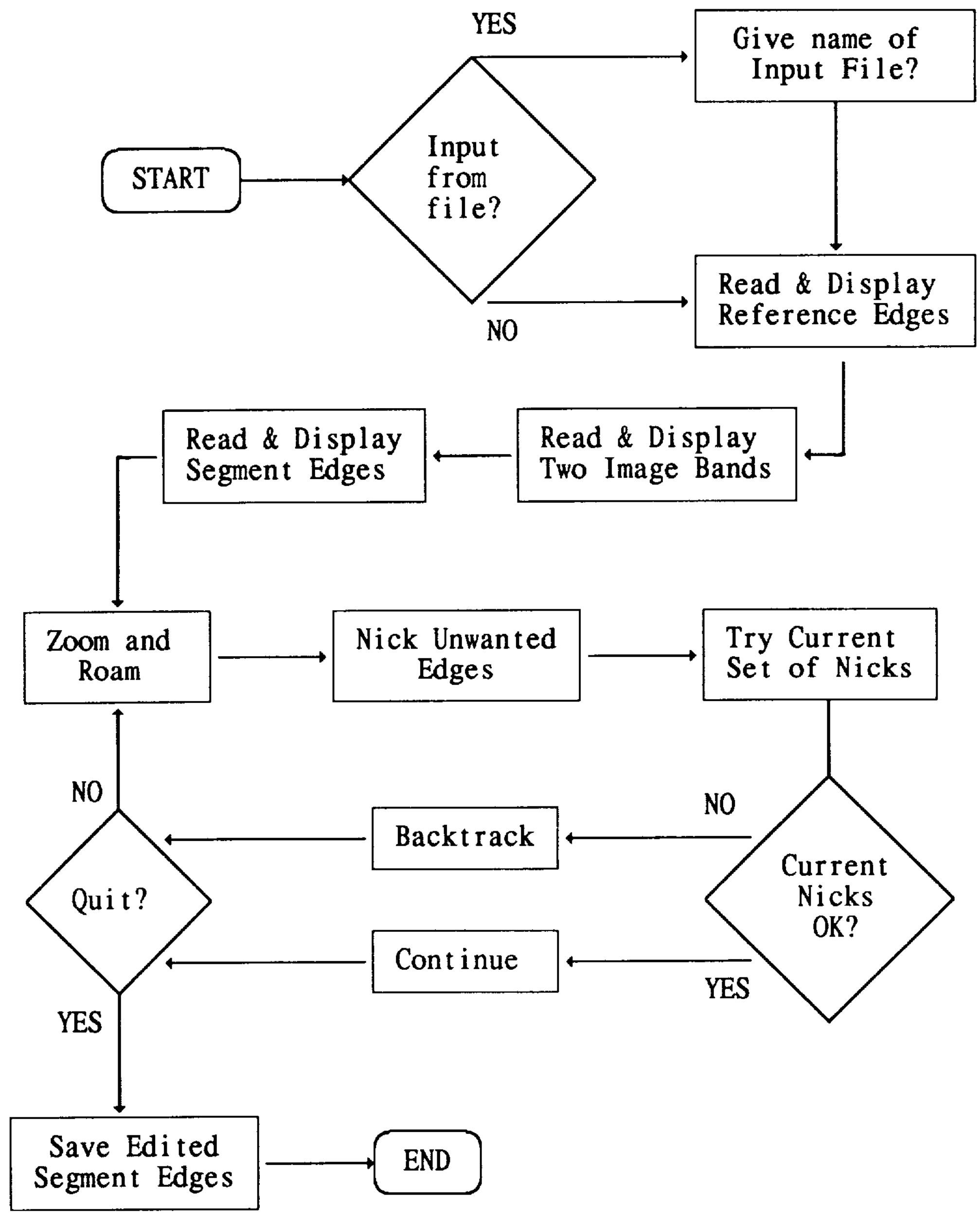

Figure 6. Flow chart outlining the steps carried out by the interactive CHGVAL#.PRO program.