EMIMGBOX.DOC

(last update Aug 17, 1987)

CONTENTS

A. INTRODUCTION

EMIMGBOX is used for boxing and floating images from the LEXIDATA screen. PLEASE BE SURE you understand exactly how this program works before using it since there are special problems associated with the use of different options.

The main concern is with the nature of the data stored in the LEXIDATA memory. In general, raw image data contains intensity values in the range 0-255 (coming from the Optronics scanning microdensitometer) and this range of values can be fully represented when the image is FIRST written to the LEXIDATA memory using the EMIMGDSP program. During boxing operations, vectors are drawn in bit plane #1 of the LEXIDATA memory to mark out the region being windowed from the rest of the image. When options C and D are used, bit plane 1 is completely cleared which forces all intensity values to be even (all intensities which initially were odd, become even after their value is decreased by one). Consequently, for options C and D the user marks out a rectangular region on the screen from which the image intensity values are read into a virtual memory array to store the original data values. The area outside this rectangle is then blacked out so the user only sees the area from which boxing can take place (i.e. the area stored in the virtual array). Now, manipulations performed on the screen are merely used to identify regions stored in memory.

In option B, vectors which outline the box are drawn non-destructively, so image intensities are maintained with their original values. Note, however, that if you have used either option C or D you are COMPLETELY OBLIGED to exit the program and redisplay the unboxed image using EMIMGDSP so the full range of image intensities are once again represented in the LEXIDATA memory (see note at end of documentation). If you started with either option A or B, you may then proceed to C or D. The user is given a warning message which, of course, can be ignored if desired (but beware of unforeseen consequences in subsequent programs which use boxed image data!).

The following list of options are currently available in EMIMGBOX:

```
                    TOUCH-KEY OPTIONS MENU
                   -----------------------
          A   Save entire 1280x1024 LEXIDATA screen image
          B   Rectangular boxing
          C   Circular boxing
          D   Arbitrary shape boxing
          E   EXIT PROGRAM
```

B. PROGRAM OPTIONS

(A) SAVE ENTIRE 1280X1024 AREA OF LEXIDATA SCREEN IMAGE

This should be self-explanatory. It will take about 45-60 seconds to write the entire screen out to a disk file (specified by the user). Please note that, with this option, the image is not floated, so the background value = 0. This option is mainly designed for saving composite images and is not used for saving images which will be processed such as by Fourier transformation, etc.

(B) RECTANGULAR BOXING

The user starts by marking the two opposite corners of the rectangular region to be boxed using the data tablet puck. After the two positions are entered, a rectangular box is drawn over the image and the user can then move and resize the box using the keypad numbers on the right side of the terminal keyboard. At this point all commands are entered in TOUCH-KEY mode (no <CR> needed). Hitting "H" or "?" will provide a synopsis of the available options at the video terminal. With a little practice the user can master the skills of moving the box around and changing its size until satisfied.

(C) CIRCULAR BOXING

The user first identifies with the tablet puck the two corners of the rectangular region over which boxing is to be performed. The larger the area chosen, the longer one must wait for the data to be stored in the virtual array. After this is finished, a circle with a default radius = 32 pixels is drawn in the center of the working window. The user then uses TOUCH-KEY commands to move the circle and change its size until satisfied. Hit either "H" or "?" to get information about the available options. You may continue to box off particles in this routine without rewriting the unboxed image as long as you don't exit back to the main program option menu and you can continue to work within the region specified when you first entered this routine.

(D) ARBITRARY SHAPED BOXING

First identify (using the tablet puck) two corners of a rectangular region over which boxing is to be performed (as is also done when option C is used). Now use the data tablet puck to mark out the vertices of the irregular shaped box. Hit the YELLOW puck button to mark new vertex positions, WHITE to close out the box (a vector is drawn between the last and first vertices marked), BLUE to erase previously drawn vectors and GREEN to EXIT from the routine and return to the main program option menu. After hitting the WHITE button, the user must position the screen cursor within the boundary of the boxed region to identify the inside of the box. A rectangular region, which just encompasses the irregular shaped box, is then written out to a disk file (filename specified by the user). The GREEN button may be used to abort a boxing session, but recall that the unboxed image may need to be rewritten to the LEXIDATA before boxing is redone.

With option D, it is undetermined what happens if boxes are drawn with vectors that cross each other.

FINAL NOTE: The reasons for setting up this program as it mainly results from considerations of execution time. The majority of boxing is done with option B which is also the simplest type. If options C and D were set up like B, the interactive manipulations (moving and changing the box size) would become excruciatingly slow. Although pre-windowing the areas to be boxed using options C and D is painful and may force redisplay of images, it seems a reasonable price to pay for speeding up the interactive manipulations. Any suggestions or comments for improvements will be gratefully acknowledged.

```
==================================================================
FORTRAN code:  DEXTRO3:[TSB.FOR]EMIMGBOX.FOR,*.S1.
Documentation: DEXTRO3:[TSB.FOR.DOC]EMIMGBOX.DOC      17-Aug-87
------------------------------------------------------------------
```