Missouri University of Science and Technology
Scholars' Mine

Opportunities for Undergraduate Research Experience Program (OURE)
Student Research & Creative Works

16 Apr 1992

Development of Software for Detecting Certain Key Features in Skin Tumor Images

Dana M. Brewer

Follow this and additional works at: https://scholarsmine.mst.edu/oure

Part of the Electrical and Computer Engineering Commons

Recommended Citation

Brewer, Dana M., "Development of Software for Detecting Certain Key Features in Skin Tumor Images" (1992). *Opportunities for Undergraduate Research Experience Program (OURE)*. 52.
https://scholarsmine.mst.edu/oure/52

This Report is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Opportunities for Undergraduate Research Experience Program (OURE) by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

DEVELOPMENT OF SOFTWARE FOR DETECTING CERTAIN KEY FEATURES IN SKIN TUMOR IMAGES

DANA M. BREWER
ELECTRICAL ENGINEERING

ABSTRACT

The goal of Dermvis is to develop a microprocessor-based system that will economically and clearly identify a skin tumor area and type. Dermvis is subdivided into various segments. This segment includes developing software, digitizing images, and manually defining borders in images. Manually drawing borders for digitized images creates bit-maps for average color analysis. Software is developed for determining the average color of the tumor and the surrounding tissue. Batch processing allows images to be analyzed in mass.

INTRODUCTION

Dermvis is the skin cancer project at the University of Missouri-Rolla with the purpose of developing a microprocessor-based system which will identify cancerous lesions. Once identified the cancerous tissue can be removed quickly.

There are seven types of skin cancers currently being considered: melanoma, dysplastic nevus, actinic keratosis (ak), squamous-cell carcinoma (scc), seborrheic keratosis (sk), basal-cell carcinoma (bcc), interdermal nevus (idn). Each genus has characteristics setting it apart from the others. High cure rates are achievable in all of these cancers if detected and treated early. Dermvis' goal is to develop software that will aid dermatologists in the early detection process.

PROJECT DEVELOPMENT

Digital Image Processing

A working image is obtained by taking pictures of the cancerous area and developing them into slides. Red, green, and blue Wratten gelatin filters and an NEC TI-23A camera are used to digitize slide images to produce bit-maps for analysis. (See Appendix C for complete equipment list).

Care should be taken to ensure accurate color depiction before digitizing. An average pixel value program, developed by

Robert McLean, is run to determine the percent difference between the colored filters while the camera is viewing the white translucent cover of the light box. A maximum of five percent difference is allowed. If the difference is greater than five percent, there is a greater risk in misidentifying the tumor class. Once acceptable color is achieved, image processing can begin.

An image is digitized by typing the following command from the keyboard.

digitize -Pc filename.pic

All image files are numbers followed by "n.pic" (e.g. 985n.pic).

The digitize program, also developed by Robert McLean, calls for focusing the image under the red filter. At the same focus, a picture of the image is taken with each filter. The result is a bit-mapped image displayed on an Androx ICS400-1A image processing board. The image focus and color is checked. If there is any question regarding the results, the necessary adjustments are made and the image re-digitized.

Each digitized image uses approximately 760 kilobytes of memory on the Solbourne 5/820 computer and file server. After storing several images temporarily in path "image/dermvis/home/dmb/images" all Dermvis group memory is depleted leaving other users without available storage space. The directory "im -> /spare200" replaced the previous path to free memory capacity for other users. The images were permanently stored on the magneto-optical read/write disk drive (eod) after write privileges were obtained.

It is important to note that all of the images stored on the eod have an identical compressed image available on the network in the directory

image/dermvis/images/compressed.

As a result of a mysterious eod failure, many hours of digitizing time were lost by failing to backup the images to network storage before deleting them from the temporary directory.

Border Identification

Part of the detection process consists of identifying the boundary of the cancerous area. A series of steps will show the bordering process.

It is first necessary to change directories to the eod to access the working images. Then, telnet from Solbourne to the PC images that need to be bordered. (Once experienced with the border program, approximately fifteen images per hour can be completed. It is recommended to border fifteen images, on average, to save time in the telnet process.)

At the C-prompt type "border". This will display a menu on the color monitor from which color images are retrieved, bordered, and saved. Robert McLean wrote this program and modified it with suggestions from the research team. The modifications improved the quality and presentation of the border program and menu.

Once points are placed around the border and splined, the area within the tumor is specified. Each byte within the border is defined as a binary one and each byte outside the border as a binary zero. The point image is saved as "filename.pnt" and the binary image is saved as "filename.bin". It is this binary image that will be used in the program that determines the average color of the tumor and surrounding skin.

After several images are bordered, telnet back to Solbourne the "*.bin" and "*.pnt" files. The asterisk includes all files with the specified suffix. To save space, all binary and point files are compressed using

```
compress *.bin
compress *.pnt
```

and stored in

```
image/dermis/border/images/compressed
```

and

```
image/dermvis/border/points/compressed,
```

respectively. To finish the bordering process, telnet back to the PC and remove all image, binary, and point files for the next user.

All bordered images are checked for accuracy by Dr. Van Stoecker, dermatologist, and discussed with the researchers. Any alterations in the tumor boundary are made and the corrected binary and point images are saved, compressed, and stored in their respective directories.

Average Color Program and Analysis

Most tumors are nevi (i.e. moles) and lesions of which color is the most identifying mark. The "Bittest" program (Appendix A) is designed to find the average color. The first section of the program makes the variable assignments. One variable, "constant unsigned character mask[8]" will be logically "anded" with the previously digitized bit-mapped image. This will identify the bytes lying within the tumor for which the color must be averaged.

The next section of the program identifies any problems in accessing files or memory. The program stops and prints the error message corresponding to the problem.

After the proper files have been opened and memory allotted, average color analysis begins. Analysis begins by finding image resolution which determines the number of bytes in the image. For each byte there are eight bits and for each bit there are three possible colors represented, one for each color plane made in the digitizing process. The program evaluates each pixel in the image and identifies it as tumor or non-tumor area. The average color of the tumor is determined by summing the color bits and dividing by the total tumor area.

"Bittest" was modified to determine the average color of the tissue defined as non-tumor. This process works the same as described in the previous paragraph. The average red in the cancerous tissue is red/area and the average red outside the tumor is redout/areaout.

The last section of the program prints the results of the analysis in a clearly identified manner.

Appendix B is a batch program designed to run a large quantity of images through the average color program. The batch program was written by Kishore Khemani and myself and is used to interface the average color program output to the color triangle program input. The statement "rm -f $output" removes any data that may be in the output file to clear it for subsequent processing. Thus, the output files are not appended to previous program runs.

The "while" loop in the batch program runs the range of the images input to the command line. It uncompresses the picture file and runs it through "Bittest", routing the output to the "set output" path. Finally, it re-compresses the picture file for storage on the network.

The second program is the color triangle developed by Kishore Khemani. This program converts the average color program output into rectangular coordinates and plots it on the color triangle. The triangle vertices represent red, green, and blue. The center of the triangle is represented by a white spot. As the software is further developed each tumor class will hopefully cluster in a specific area on the color triangle.

AREAS FOR FURTHER CONSIDERATION

It is important to back up the work. Many hours of digitizing could have been saved by compressing the images and storing them on the network prior to writing them to the eod. Eod problems and failures were investigated on several occasions and a root cause has yet to be found. Cost considerations make replacement of the eod unlikely.

It may be appropriate to consider a heat analysis of the cancerous area. A comparison of the heat differential between the cancerous area and non-cancerous area may aid in the border definition process. This would reduce the amount of time taken to manually define and check tumor boundaries. This type of analysis however, will take more funding and equipment than is presently available and is not feasible without outside sponsoring.

Another research possibility as an aid in border definition is use of magnetic resonance imaging (MRI). MRI applies strong magnetic fields and radio-frequency pulses to the soft tissues of the body. The image is transmitted to a monitor from which the tissues may be analyzed for border, heat, luminescence, etc. This process is also costly, but may be possible with cooperation from the Phelps County Regional Hospital.

CONCLUSION

There is much frustration with the equipment. Without the eod working properly it is very difficult to acquire image files to test software development. The average color program works and all of the images have been bordered, except those most recently digitized.

Acknowledgments

Dr. Randy H. Moss, faculty advisor, provided me the opportunity to participate in the Dermvis research project.

Dr. W. Van Stoecker, dermatologist, initiated Dermvis and invited student research participation.

APPENDIX A
BITTEST

```
#include <stdio.h>
main(argc, argv)
int argc;
char *argv[];
{

  /*variable assignments*/

  FILE *image, *color;
  int i, xres, yres, done;

        /*color values for tumor*/
  long red, green, blue, area;

        /*color values for area outside the tumor*/
  long redout, greenout, blueout, areaout;

  unsigned char  tumor, r, g, b;
  const unsigned char mask[8]={128, 64, 32, 16, 8, 4, 2, 1};
  char *magic;

  /*error message for accessing program*/

  if (argc!=3)
  {
    fprintf(stderr,  "USAGE: %s bitfile picfile\n", argv[0]);
    return -1;
  }

  /*error message for accessing the image file*/

  image= fopen(argv[1], "r");
  if (image==NULL)
  {
      fprintf(stderr, "Something went wrong when opening image
file.\n");
    return -1;
  }

  /*error message for memory assignment*/

  magic= (char *) malloc(6);
  if (magic==NULL)
  {
       fprintf(stderr,  "Something went wrong when allocating
memory.\n");
    return -1;
  }
```

```
  /*error message for file type*/

  for (i=0; i<5; i++)
    magic[i]= (char) getc(image);
  magic[i]='\0';
  if (strcmp("FBMAP", magic)!=0)
  {
    fprintf(stderr, "This is not a bit map file.\n");
    return -1;
  }

  /*error message for accessing image file*/
  color= fopen(argv[2], "r");
  if (color==NULL)
  {
      fprintf(stderr,  "Something  went  wrong  when  opening  color
file.\n");
    return -1;
  }

   /*Once  the  other  tests  have  been  passed  the  average  color
analysis begins.*/

  printf("This is a bit map file.\n\n");
  xres=getc(image);
  xres+=getc(image)<<8;
  yres=getc(image);
  yres+=getc(image)<<8;
  done=xres*yres;

        /*"done" is the number of bytes on the screen*/
  done=done/8;

        /*initializing tumor shades to zero*/
  red=green=blue=0;

        /*initializing area of tumor to zero*/
  area=0;

        /*initialize nontumor shades to zero*/
  redout=greenout=blueout=0;

        /*initialize nontumor area to zero*/
  areaout=0;

/*One time through the loop reads value of each color bit in the
byte for a total of "done" loops*/

  while(done>=0)
```

```
  {
    tumor=getc(image);
    done--;
    for(i=0; i<8; i++)
    {
      r=getc(color);
      g=getc(color);
      b=getc(color);

/*Find average color of the tumor tissue.*/

      if((tumor&mask[i])!=0)
      {
        red+=r;
        green+=g;
        blue+=b;
        area++;
      }

        /*Find the average color of the tissue surrounding the
tumor.*/

      else if(r>1.05*g && r>1.05*b && r>50)
      {
        redout+=r;
        greenout+=g;
        blueout+=b;
        areaout++;
      }
    }
  }
  printf("Average tissue color within the tumor for\n");
  printf("%s.\n\n", argv[1]);
  printf("Red %9.3f\n",(double) red/area);
  printf("Green %7.3f\n",(double) green/area);
  printf("Blue %8.3f\n\n",(double) blue/area);
  printf("Average tissue color outside the tumor.\n\n");
  printf("Red %9.3f\n", (double) redout/areaout);
  printf("Green %7.3f\n", (double) greenout/areaout);
  printf("Blue %8.3f\n", (double)blueout/areaout);
}
```

APPENDIX B
PROGRAM FOR BATCH PROCESSING

```
#! /bin/csh

#set any and all of the variables that will be used in this
#program

set output = /image/dermvis/home/dmb/src/skin/output
set image_dir = /eod
set bin_img_dir = /image/dermvis/border/images/compressed
set program = /image/dermvis/home/dmb/src/skin/bittest

#make sure the proper number of parameters were passed on the
#command line

if($#argv != 2) then
  echo "USAGE:  first_img last_img"
  exit -1
endif

#switch order of parameters if they were given in the wrong order

if ($1 > $2) then
  @ first = $2
  @ last  = $1
else
  @ first = $1
  @ last  = $2
endif

#start the batch

#remove the previous results

rm -r $output

#create a new result

while($first <= $last)
  if(-e $bin_img_dir"/"$first".bin.Z") then
    uncompress $bin_img_dir"/"$first".bin.Z"
                          $program     $bin_img_dir"/"$first".bin"
$image_dir"/"$first"n.pic" >> $output
    compress $bin_img_dir"/"$first".bin"
  endif
  @ first++
end
```

APPENDIX C
EQUIPMENT

Androx ICS400-1A image processing board
Solbourne 5/820 computer and file server
TrueVision ATVista 2M color frame-grabber board
20 MHz 80386-based AT-compatible computer
IDEK MF5015 color multi-sync monitor
NEC TI-23A camera
32 Mbyte RAM and 2 1024 Mbyte hard drives
Magneto-optical read/write disk drive (eod)
Red, green, and blue Wratten gelatin filters