

Tutorial/Lab Session 10

PURPOSE:

1. To experiment with the detection of boundary pixels (after binarization) by using a “boundary tracing” algorithm.

PROCEDURE:

Practice 1: To prepare for doing the detection of boundary pixels

Step 1: Login to the PC and start the X-window environment.

Step 2: Go to the directory xvision (use “cd xvision”).

Step 3: Edit the file “xvision.c” with “emacs” (use “emacs xvision.c”).

Step 4: Add in the following contents before and inside the

function Test8():

```
void RecordBoundaryPixel(int r, int c, int label)
{
}
void TraceBoundary(int label)
{
}
void Test8()
{
    char filename[80];
    int r, c, label ;
    labelmap = image_out ;
    printf("\n> Enter image file :");
    scanf("%s", filename) ;
    ReadImage(filename) ;
    DrawPixmap(oneimage, 256, 256, 0, 0) ;
    /* Binarize image */
    BinarizeImage(oneimage) ;
    /* Group pixels and Trace boundary */
}
```

Step 5: Compile the program (type “make”).

Step 6: Execute the program (type “xvision”) and click on the button <Test 8>.

Step 7: Enter the image file: image4.img and identify a good threshold value.

Practice 2: To label image regions (same as in Tutorial/Lab Session 9)

Step 1: Add in the following content into the function Test8():

```
void Test8()
{
    (same as in Practice 1)

    /* Group pixels and Trace boundary */
    label = 20 ;
    for (r = 0 ; r < 256*256; r++)
    {
        labelmap[r] = 0 ;
        image1[r] = 1; /* image of doundary */
    }
    for (r = 0 ; r < 256 ; r++)
        for (c = 0 ; c < 256 ; c++)
            if (oneimage[r*256+c] == 1)
            {
                AssignLabelToPixel(r, c, label) ;
                TraceBoundary(label) ;
                label = label + 20 ;
                if (label > 255) label = 15 ;
            }
    DrawPixmap(image1, 256*256, 0, 256) ;
    DrawPixmap(labelmap, 256*256, 256, 256);
}
```

Step 3: Compile the program (type “make”).

Step 4: Execute the program (type “xvision”) and click on the button <Test 8>.

Step 5: Enter the image file: image4.img and a good threshold value.

Step 6: Observe the results.

Practice 3: To trace the boundary of labeled image regions.

Step 1: Develop the function TraceBoundary() with the following content:

```
void TraceBoundary(int label)
{
    int r, c;
    for (r = 0 ; r < 256 ; r++)
        for (c = 0 ; c < 256 ; c++)
            if (labelmap[r*256+c] == label)
                {
                    RecordBoundaryPixel(r, c, label);
                }
}
```

Step 2: Add in the following contents before and inside the function RecordBoundaryPixel():

```
static int IsBoundaryPixel(int r, int c, int label)
{
    if (labelmap[(r-1)*256+c] != label) return 1;
    if (labelmap[(r+1)*256+c] != label) return 1;
    if (labelmap[r*256+c-1] != label) return 1;
    if (labelmap[r*256+c+1] != label) return 1;
    return 0;
}
void RecordBoundaryPixel(int r, int c, int label)
{
    if (image1[r*256+c] == label) return ;
    image[r*256+c] = label;
    if (IsBoundary(r-1, c-1, label) == 1)
        RecordBoundaryPixel(r-1, c-1, label);
    if (IsBoundary(r-1, c, label) == 1)
        RecordBoundaryPixel(r-1, c, label);
    if (IsBoundary(r-1, c+1, label) == 1)
        RecordBoundaryPixel(r-1, c+1, label);
    if (IsBoundary(r, c-1, label) == 1)
        RecordBoundaryPixel(r, c-1, label);
    if (IsBoundary(r, c, label) == 1)
        RecordBoundaryPixel(r, c, label);
    if (IsBoundary(r, c+1, label) == 1)
        RecordBoundaryPixel(r, c+1, label);
    if (IsBoundary(r+1, c-1, label) == 1)
        RecordBoundaryPixel(r+1, c-1, label);
    if (IsBoundary(r+1, c, label) == 1)
        RecordBoundaryPixel(r+1, c, label);
    if (IsBoundary(r+1, c+1, label) == 1)
        RecordBoundaryPixel(r+1, c+1, label);
}
```

(continued to the next page)

(from previous page)

Step 3: Compile the program (type “make”).

Step 4: Execute the program (type “xvision”) and click on the
button <Test 8>.

Step 5: Enter the image file: image4.img and a good threshold value.

Step 6: Observe the results.

CREATIVE WORK

To visually monitor the execution of the “boundary tracing” program.

“xvision” provides a graphic function called DrawRectangle() to display a rectangle around a point (x, y) (for example, DrawRectangle(x-2, y-2, x+2, y+2, 0)). You can use this function to highlight the boundary pixels during the execution of the program. The idea is simple: just draw a rectangle around the newly identified boundary pixel.

Implement this idea and observe the behavior of your program.