Introduction to Computer Vision
Lab - 4
Region Interaction

Submitted By:
Saroj Kumar Dash

October 24, 2017

Contents

1 Problem Statement

To Implement region growing using interactive GUI. I build my code based upon the two code stubs given in class.
Given Codes in class:

1. region_grow.c code with respect to intensity
2. win32 program to make the GUI to interact with the image

I initially started with designing all the UI. I created the UI elements of the system. After making the UI elements I started to link the UI elements to the region grow code to make it interactive in nature on the image that we load. The problem statement had many UI elements to add. In the following section I will explain all the UI elements that I have implemented on my side along with the code.

2 Solution

2.1 Solution element 1:

The program should visualize the region growing by coloring pixels as they join the growing region. The program should have a GUI option that allows the user to select the color for pixels that join the region.

My Solution:

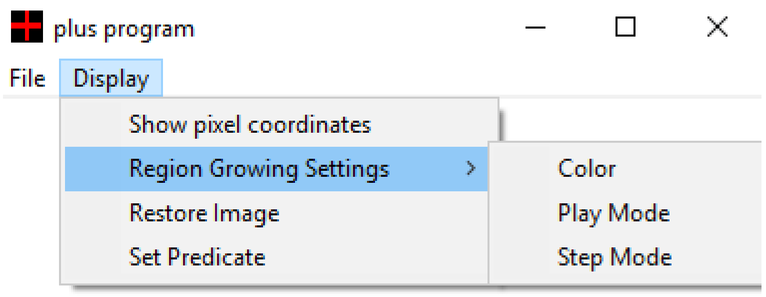
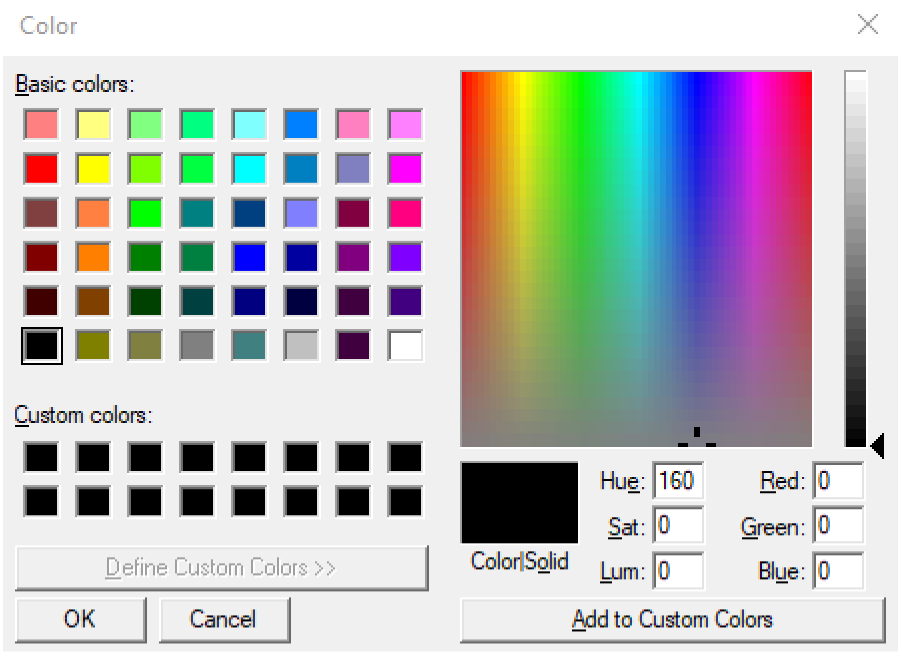


Figure 1: All menu options

1. Region Growing Settings → Color : the user can use this to set color by selecting from the color pallette that opens after this button is clicked.

2. Opens up the below color pallette to select the pixel color:



Hue: 160
Sat: 0
Lum: 0
Red: 0
Green: 0
Blue: 0

Figure 2: Color Pallette

3. Restore the image or clearing the image. Use the menu option Restore image to restore the image.

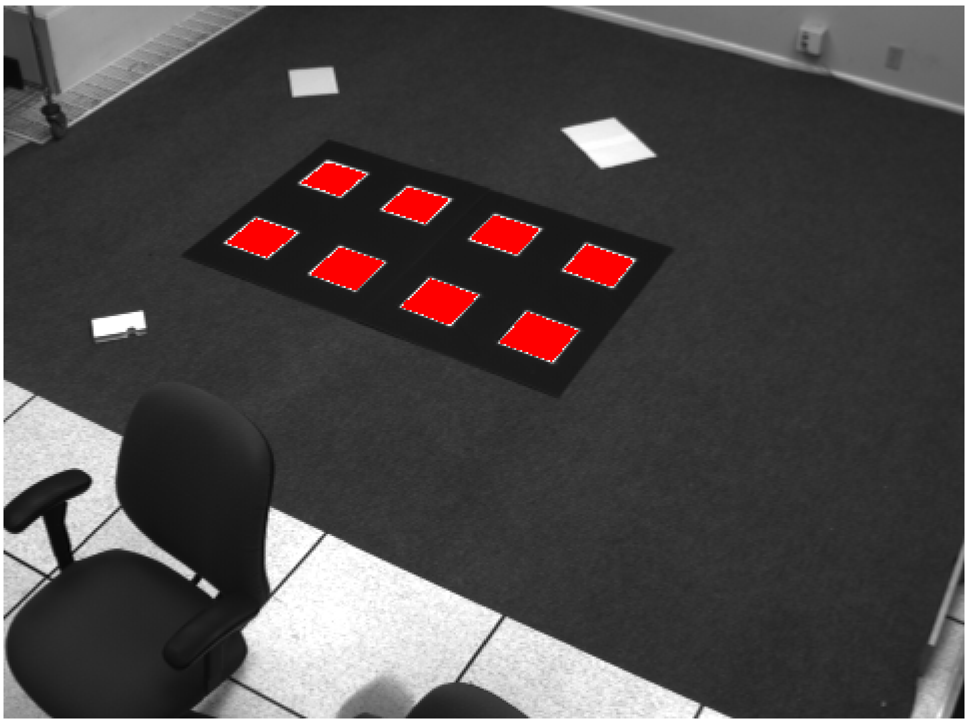


Figure 3: Before Restore Image

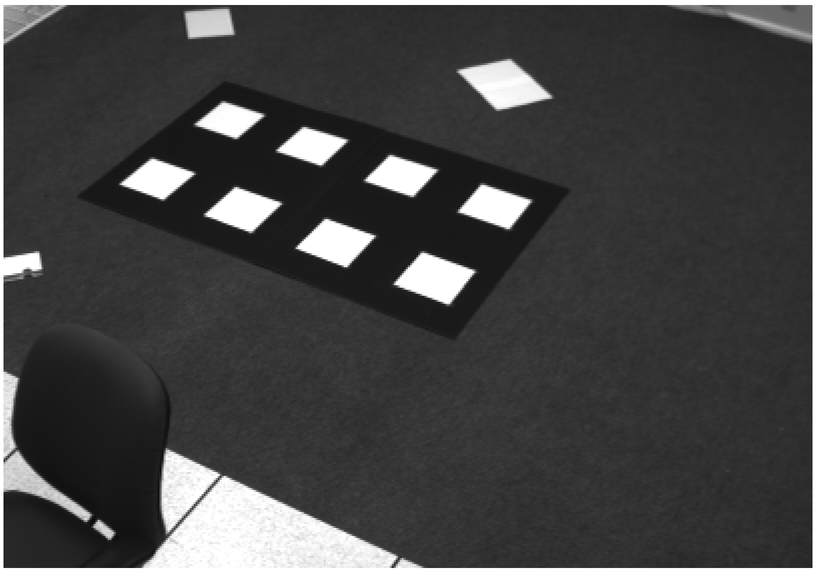


Figure 4: After Restore Image

2.2 Solution Element 2:

The program should have a menu option to grow the region in play mode or in step mode. In play mode, a pixel should join the region each 1 ms. In step mode, a pixel should join the region each time the user presses the key j. The program should allow the user to change between modes while a region is growing.

My Solution: As shown above in the menu options. the Region grow settings has two options to set playMode and StepMode. The modes are handled in there callback methods as below:

```
case ID_REGIONGROWINGSETTINGS_PLAYMODE:
        jmpMode = 0;
              playMode = (playMode + 1) % 2;
              bwait = 0;
              image_reloaded = 0;
        break;
case ID_REGIONGROWINGSETTINGS_STEPMODE:
              playMode = 0;
              jmpEvt = CreateEvent(
                                    NULL,
                                    FALSE,
                                    FALSE,
                                    TEXT("J-Event")
                       );
              jmpMode = (jmpMode + 1) % 2;
              image_reloaded = 0;
              break;
```

As we can see from the above code when the region growing is happening in play mode then if we press j then the systems changes to step mode and then stops its play loop, to work upon only j event mode.

Image after play mode with default distance and intensity which is 500 and 100 respectively.

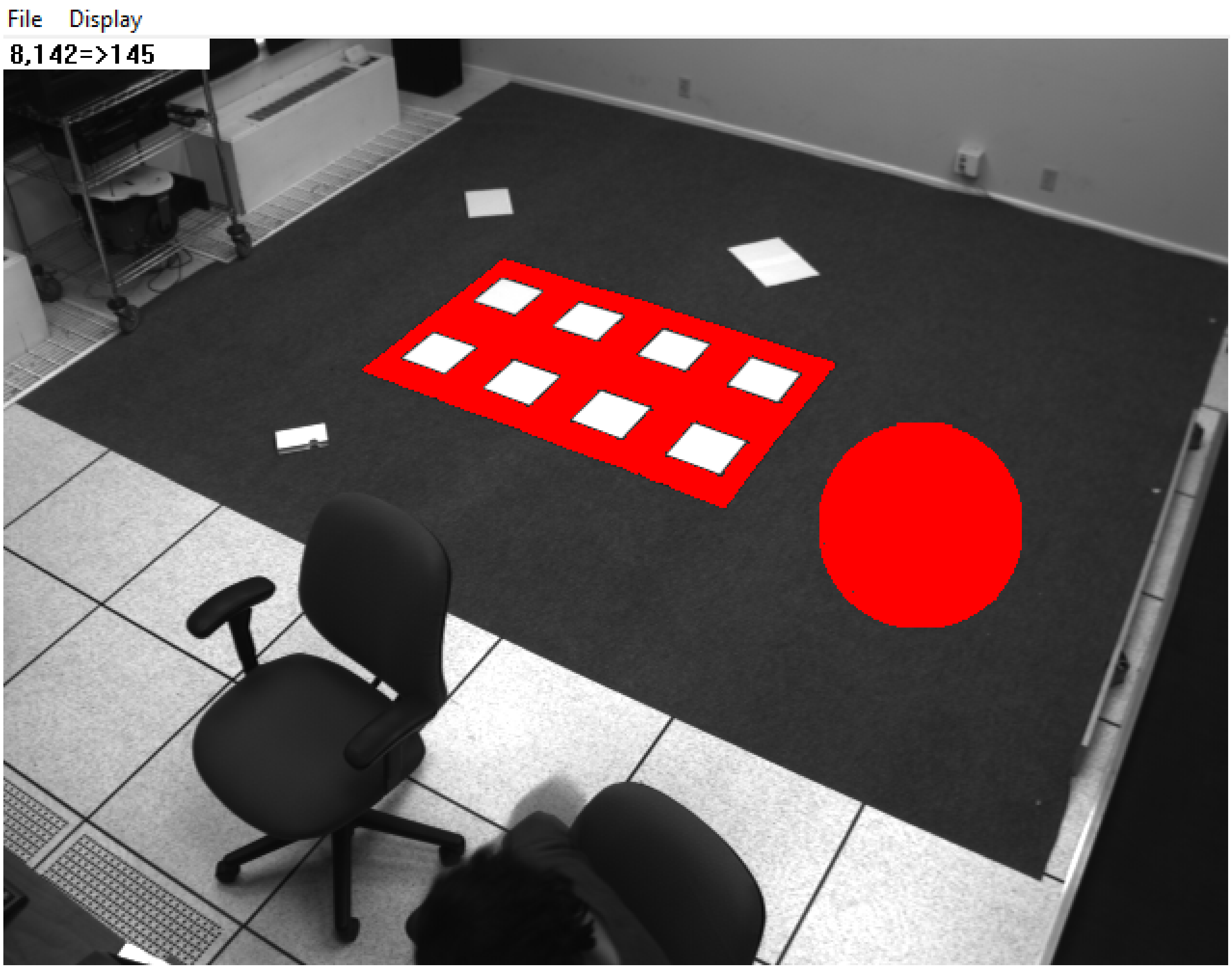


Figure 5: Region Growing using playMode interaction

Then we can use **Restore Image** button to again restore the image back to original version. We use the below code to restore the image to original form.

```
for (int i = 0; i < ROWS*COLS; i++)
        image_copy[i] = OriginalImage[i];
      PaintImage();
```

2.3 Solution Element 3:

The program should use a dialog box to allow the user to select values for two predicates for joining the region. The first predicate is the absolute difference of the pixel intensity to the average intensity of pixels already in the region. (To join, a pixel must be within this range.) The second predicate is the distance of the pixel to the centroid of pixels already in the region. (To join, a pixel must be within this range.) Both predicates should be applied at the same time while growing a region.

My Solution: I made a dialog element, using the standard dialog box building tools interface.rc and the dialog element is activated by the below option: Display → Set Predicate.

The dialog that opens appears below:
The distance and intensity is taken into picture in the function region. This is also called as the predicate region rules. A pixel is shaded red only if its difference between its own intensity or pixel value from the avg intensity value is less than or equal to the intensity taken from the dialog box. After the

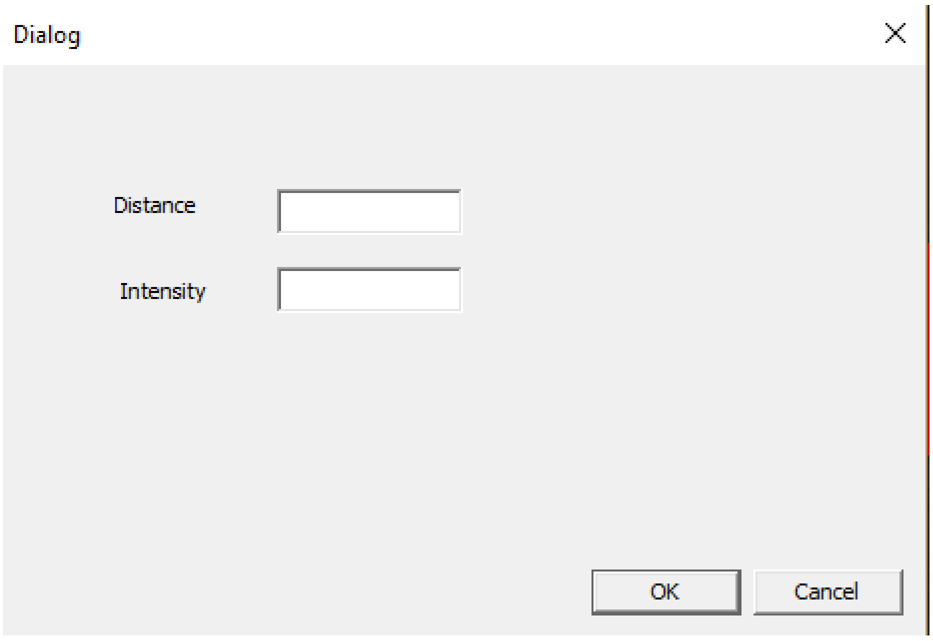


Figure 6: Dialog for distance and Intensity setting

pixel matches the intensity criteria then we look for the distance from the centroid of the pixel that is chosen and if the distance is less than the taken value from the dialog then it is shaded as red.

The rule that I added to check the intensity rule:

```
//the intensity rule of predicate
diff = abs((int)(image_copy[(queue[qt] / COLS + r2)*COLS + queue[qt] % COLS + c2]) - avg
if ((diff) > intensity)
        continue;
```

The rule that I added in the regionGrow code to get the distance rule is :-

```
//euclidean distance
abs_dist = (int)abs(sqrt(SQR(xnbr - queue[qt] / COLS + r2) + SQR(ynbr - queue[qt] % COLS

if (distance != 0)
//the distance rule of the predicate
if (abs_dist > distance)
        continue;
```

3 Appendix

```
void color_image(int *local, int count)
{
        int x = 0, y = 0;
        int i = 0;
        HDC hdc = GetDC(MainWnd);
        if (count == 0)
                goto EXIT;

        for (i = 0; i < count; i++)
        {
                x = local[i] / COLS;
                y = local[i] % COLS;
                SetPixel(hdc, y, x, RGB(GetRValue(new_color), GetGValue(new_color),GetB
        }

EXIT:
ReleaseDC(MainWnd, hdc);
return;
}

//This is how we captutre the distance and intensity value for the predicate value in th
case ID_SET_PREDICATE:
                    ret = DialogBox(GetModuleHandle(NULL), MAKEINTRESOURCE(IDD_DIALOG1), h
                    PaintImage();
                    break;

BOOL CALLBACK getPredicate(HWND hWnd, UINT Message, WPARAM wParam, LPARAM lParam)
{
        int local_dist, local_intensity;
        BOOL ferr1, ferr2;
        switch (Message)
        {
        case WM_INITDIALOG:
                return TRUE;
        case WM_COMMAND:
                switch (LOWORD(wParam))
                {
                case IDOK:
                        local_dist = GetDlgItemInt(hWnd, IDC_EDIT1, &ferr1, FALSE);
                        local_intensity = GetDlgItemInt(hWnd, IDC_EDIT2, &ferr2, FALSE);
                        if (!ferr1)
                        {
                                if (!ferr2)
                                {
                                        MessageBox(hWnd, "No values entered for Distance
                                }
                                else
                                {
                                        if (local_intensity < 0 || local_intensity > 25
                                                MessageBox(hWnd, "Please enter correct v
                                        else
                                                intensity = local_intensity;
                                }
                        }
                        else
                        {
```

```
                                        if (local_dist< 0)
                                                MessageBox(hWnd, "Please enter correct values fo
                                        else
                                                distance = dist;
                                        if (ferr2)
                                                if (local_intensity < 0 || local_intensity > 255
                                                        MessageBox(hWnd, "Please enter correct v
                                                else
                                                        intensity = local_intensity;
                                }
                                EndDialog(hWnd, IDOK);
                                break;
                        case IDCANCEL:
                                EndDialog(hWnd, IDCANCEL);
                                DestroyWindow(hWnd);
                                return TRUE;
                                break;

                        }
                        break;
                default:
                        return FALSE;
                }
                return TRUE;
        }

        //applying the regiongrow logic when the mouse is clicked at a point in image in the mai
        case WM_LBUTTONDOWN: case WM_RBUTTONDOWN:
                c = LOWORD(lParam);
                r = HIWORD(lParam);
                if (c >= 0 && c < COLS  &&  r >= 0 && r < ROWS)
                {
                        hDC = GetDC(MainWnd);
                        paint_over_label = GetRValue(GetPixel(hDC, c, r));  //gets the last va
            //paint_over_label = GetGValue(GetPixel(hDC, c, r));
            //paint_over_label = GetBValue(GetPixel(hDC, c, r));

                        new_color = rgb_curr;
                        new_label = -1;
                        ReleaseDC(MainWnd, hDC);
                        _beginthread(RegionGrow, 0, MainWnd);
                }
                return(DefWindowProc(hWnd, uMsg, wParam, lParam));
                break;
        //end of the case of handling the
```