The Model For Extracting A Portion Of A Given Image Using Color Processing

Hardik Pandit
G H Patel Post Graduate Department of
Computer Science, Sardar Patel University
Near Jain Derasar, Nana Bazar,
Vallabh Vidyanagar, Gujarat, India.

Dr. D M Shah
G H Patel Post Graduate Department of
Computer Science, Sardar Patel University
Near Jain Derasar, Nana Bazar,
Vallabh Vidyanagar, Gujarat, India.

Abstract:
In this paper the model for extraction of an interested portion of an image for further processing is discussed and presented by taking example of the medical palmistry. Medical palmistry is a science of observing nails and palm to predict some disease. In Medical science also, color of palm and nails contribute in prediction of diseases like jaundice. This paper presents work done to get color of palm from the palm image and it works successfully on the different skin tones of human palm. The model increases accuracy of such observations of palm.

Introduction:
The color of palm and nail is observed carefully by many doctors to get assistance in disease identification. It is possible to observe color of palm and nail by naked eyes, but it may become subjective. Computer vision helps us to determine this color without any subjectivity. The model scans the palm through scanner, and then digital image processing and analysis techniques are applied to get color information. While doing this, first the portion of the palm should be fetched from the scanned image and then, palm color is to be identified, which may vary in different regions of palm. Following model is designed for the same purpose. The model is implemented using ASP.Net with C#. AForge.Net is used to get additional image processing filters.

System Model:

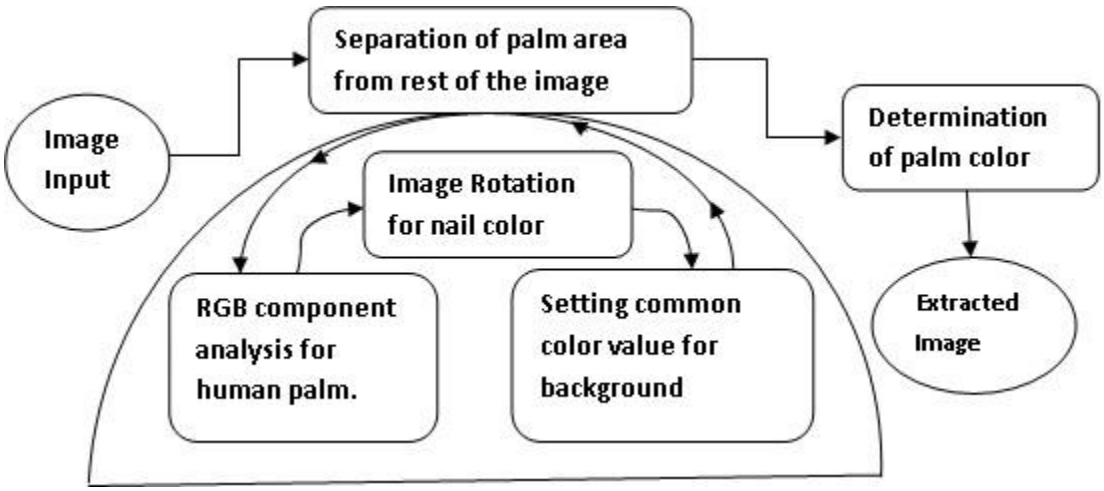

Figure 1:Model for extracting a portion of a given image using color processing

Separation of palm area from rest of the image
To start the process, first scan the palms through flat-bed scanner. Each palm is scanned from both the sides, i.e. front side and back side. Front sides of both palms would help us to get color of palms and back

side images would give us skin tone as well as nail color. Following steps are followed to separate area of palm from rest of the image.

Step 1: Careful observations of RGB color values of different skin tones of human show that, in the image of human skin, red component is having higher value than green and blue components. This observation sets logic behind this step. List of image points is used to store all those pixels that satisfy the above condition. The C# code segment is as follows:

```
if ((oimg.GetPixel(x, y).R > (oimg.GetPixel(x, y).G) + 10 ) && (oimg.GetPixel(x, y).R > (oimg.GetPixel(x, y).B + 20 )))
{
//Selectiong only those pixels which are in the area of palm and adding it to //the list
imgpoints.Add(new IntPoint(x, y));
}
```

Here, oimg is original bitmap image that is being processed. But as it is well known that GetPixel() method is very slow, the processing was very time consuming. Especially, when we have to process pixel by pixel, and each pixel may have 24 bits as color information, this method is not tolerable. To reduce processing time, byte pointer has been used as below:

```
UnmanagedImage img = new UnmanagedImage(imageData);
int pixelSize = (img.PixelFormat ==     System.Drawing.Imaging.PixelFormat.Format24bppRgb) ? 3 : 1;
byte* p = (byte*)img.ImageData.ToPointer();
int totpix = height * width;
// for each line
for (int y = 0; y < height; y++)
{
   // for each pixel
   for (int x = 0; x < width; x++, p += pixelSize)
   {
    … code that uses GetPixel(x,y);
    }
 }
```


The code is working for different tones of skin. The results for two extremely different skin tones are shown in figure 1. One can see in figure 1 (b) that given code does not include cuff of shirt. It only focuses on skin tone.

If during scanning of palms, there are any other objects on scanner, or user has worn ring or bracelet, the system does not include that region for calculating average color of palm, as shown in figure 2. In this case, of course, accuracy of system reduces a little. Yet advice has been given to our users not to wear rings, bracelets, or full sleeve clothes which may reduce accuracy of the system.

Step 2: As far as color processing of palm is concerned, any inclination of palm gives same result. For nail's color processing vertical position of palm in image is needed. User can scan his/her palms either horizontally or vertically. Horizontally scanned palm is automatically rotated and made vertical by the system. The logic behind this step is that, any human palm will have length greater than breadth. Code segment for this is as below:

```
IntPoint maxy = new IntPoint(1,1); //initializing the bottom point
IntPoint miny = new IntPoint(1000,1000);//initializing the top point
IntPoint maxx = new IntPoint(1,1); //initializing the rightmost point
IntPoint minx = new IntPoint(1000,1000);//initializing the leftmost point

foreach (IntPoint i in imgpoints)
        {
```

```
    //getting maximum and minimum coordinate values
    if (i.Y > maxy.Y)
    {
       maxy = i;
    }
    if (i.Y < miny.Y)
    {
       miny = i;
    }
    if (i.X > maxx.X)
    {
       maxx = i;
    }
    if (i.X < minx.X)
    {
       minx = i;
    }
}

//--> Rotating the image, if it is horizontal
if ( (maxy.Y- miny.Y) < (maxx.X - minx.X))
{

  //create filter - rotate for 270 degrees keeping original
                //image size
                  RotateNearestNeighbor filter = new          RotateNearestNeighbor(270);
  // apply the filter
   oimg = filter.Apply(oimg);
}
```

The result of this code segment is shown in figure 3.

Step 3: After adding selected points in this list, all the remaining pixels are set with common new color values using SetPixel() function , so it would be easy to eliminate them in further processing of color of palm.

In this way both the palms in the image are separated from background.

Determination of palm color

Once the area of palm is determined, one get the RGB color value for each pixel by using GetPixel() method. The selected pixels are stored in a list. GetPixel() method is applied to get color value of each pixel in this list. Then arithmetic means of red, green, and blue color components are calculated as follows:

```
foreach (IntPoint i in imgpoints)
    {
       oimg.GetPixel(i.X, i.Y));
       //separating values of each component
       int x1 = oimg.GetPixel(i.X,i.Y).R;
       int y1 = oimg.GetPixel(i.X, i.Y).G;
       int z1 = oimg.GetPixel(i.X,i.Y).B;

       //adding value of each component to its respective variable
       totalred += x1;
       totalgreen += y1;
       totalblue += z1;
       count += 1;

       //averaging the values
```

```
    avgred = totalred / count;
    avggreen = totalgreen / count;
    avgblue = totalblue / count;
  }
```

Using this code, average RGB color value of palms is calculated.

Cropping the region of interest

After extracting the palm region, the rectangle containing this area is cropped. Because of this, further processing is limited to this rectangle only, which saves time and reduces chances of errors. This is shown in figure 1 (b).

Limitation

In the scanned palm image, the borders of palm are darker than palm color. So, one may not get exact boundary of palm. This can effect in the calculation of average color of palm. Image enhancement methods can be used to overcome this problem. Moreover, if the surface of scanner is spoiled with stains, then also results may vary.

Conclusion

The model accurately extracts the palm portion and gives average color of human palm. These values would be used to predict diseases on the basis of medical palmistry. The model is working successfully for different skin tones of human palms.

Results

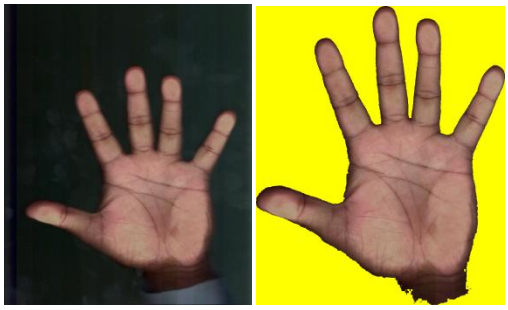

Figure 1: (a) is input image and (b) is resultant image respectively from left to right.

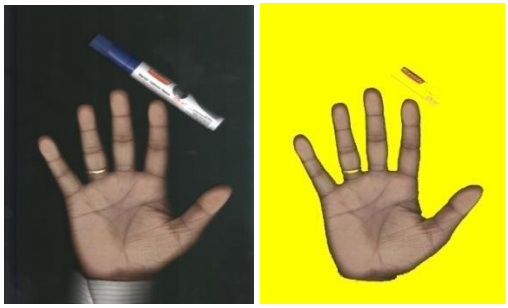

Figure 2: (a) is input image and (b) is resultant image respectively from left to right.

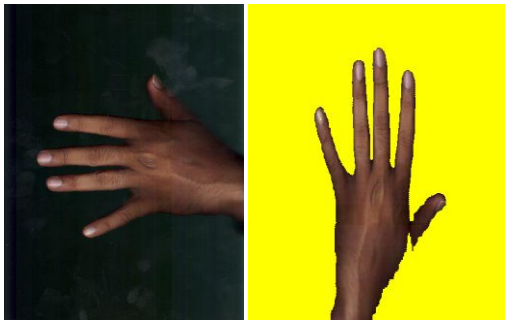

Figure 3: (a) is input image and (b) is resultant image respectively from left to right.

References

[1] R. C. Gonzalez and R. E. Woods "Digital Image Processing", 2nd edition, Pearson Education, 2004
[2] H. B. Pandit and D.M.Shah "Decision Support System for Medical Palmistry" in "Advances in Applied Research", vol.2, July-December 2010, pp 173-178.