Examining False-Colour schema in difference images

Image Processing Assignment One

Nicholas Chop

University of Guelph

The use of false colour images is common in many image processing tasks. False colour images are created from gray level or spectral images by mapping a colour value onto a gray level. Usually a lookup table is used to find the mapping. However, a static look up table can sometimes be ill-suited to representing an image. Several adaptive colour schemes are proposed and evaluated using difference images created from medical images.

Image differencing is a simple way of comparing two images of the same size and gray level depth. Each pixel in the output image is set to the first image's gray level minus the second image's gray level. Note that for a 255 gray level image, the output can vary between 255 and 255. Thus, to display the difference image the output must often be normalized. Performing image differencing in this way produces a relative difference: features which are bright in image 1 and dark in image 2 will have high normalized output values. Conversely, features which are bright in image 2 but dark in image 1 will have low normalized output values. Features with similar intensities in both images will have a midrange normalized output value. Difference images typically have a spike of pixels around the zero, or "no difference" value, but meaningful features can appear anywhere inside the range of output values. It is also important to note that in a normalized difference image, the "no difference" value does not necessarily appear at the midpoint of the gray level range.

A variation on image differencing is to use the absolute value of the difference between the pixels, producing a measure of absolute difference between the two images. This form of difference image highlights differences without being concerned with where in the two images the difference occurred. Absolute difference images tend to have a large proportion of their histogram in the low-valued area, but as with relative difference images important features can occur anywhere inside the gray level space.

The technique of false colour imaging is widely used in a variety of image processing tasks. The human visual system can only distinguish a few dozen gray levels, but is capable of separating several thousand colour hues. By converting a gray level image into a colour image, details in the image become much more apparent to the human visual system. Essentially, producing a false colour image enhances the contrast of features within an image.

False colour is typically applied to a gray level image through the use of a lookup table. Each gray level is assigned a corresponding colour. The new image is generated by examining the gray level image pixel by pixel, finding the gray level for a pixel, and assigning the corresponding colour from the lookup table into the colour image. This is a rapid algorithm, since for each pixel in the image only one lookup needs to be performed.

Most existing false colour algorithms use a static, predefined look up table. These typically assign colours evenly across the whole colour space. Although contrast is improved, in certain images much of the colour space is assigned to sparse areas in the gray level histogram. There is also no method for locating features in the image, such as background or foreground, and assigning them appropriate colours.

This paper proposes several false colour schema designed to produce better contrast between features inside a gray level difference image. Each of these was chosen in an attempt to maximise contrast between features inside a gray level difference image.

The first algorithms implemented were simple colour spectra for use as a base comparison. For the RGB spectra, the gray level space was divided into two sections. For the first half of the gray space, the red spectral component was decreased from 255 to 0 in even steps, while the green component was increased from 0 to 255 in even steps. For the second half of the gray space, the green component was decreased from 0 to 255, while the blue component was increased from 0 to 255. This produced an even spectrum from red through green to blue. The HSI spectrum was generated by setting the saturation and intensity components to 1, while rotating the hue component from 0 to 360 degrees in even steps. Due to the properties of the HSI space, this produced a spectrum starting at red, going to green, then blue, then back to red. Although these colour spaces work, they don't lend themselves to dynamic assignment well. Since the lower and upper bounds on the colour space always need to have colour values anchored, the RGB space has only the green point as an algorithm-specifiable control point. The HSI space has the green and blue points specifiable, but has a disadvantage in that the lower and upper values in the colour space both approach red. For images with broad ranges, this can cause perceptual confusion between lower and upper values.

In order to correct these problems, the simple spectra were extended to include more colour information and more available control points. The RGB spectrum was extended twice. First, it was extended by adding two more colour transitions, so the colour progression became black-red-green-blue-white. The second RGB extension mixed up the colour progression, becoming white-red-black-green-blue. The HSI spectrum was extended by adding a progression from intensity 0 to 1, with saturation at 1 and hue at 0. Then hue was rotated through 240 degrees. Finally, the hue was rotated from 240 to 360 degrees while saturation was decreased from 1 to 0.5. This produced a black-red-green-blue-ochre colour progression. All three of these colour spectra added more colour depth to the look up table, and also increased the number of algorithm-specifiable control points to three.

Three different ways of dynamic colour space allocation are proposed. The first, mean-centered histogram partitioning, attempts to place the midpoint of the colour space at the mean of the gray levels in the image. The other two control points are set halfway between the mean and their respective end values. Pseudocode for the mean-seeking algorithm follows:

```
sumHistogram = 0;
for (i = 0; i < greyDepth; i++)
{
  sumHistogram = sumHistogram + (histogram[i]*i);
}

//now we have the mean greylevel
meanHistogram = sumHistogram/greyDepth;

//need to find the histogram index closest to the mean
meanIndex = 0;
minimumError = maximumHistogram; //or some other huge value
for (i = 0; i < greyDepth; i++)
{
  error =absolute(histogram[i]meanHistogram);

  if (error < smallestError)
  {
    smallestError = error;
    meanIndex = i;
  }
}

//now set the 3 control points for the colour spectrum

firstPoint = meanIndex/2;
secondPoint = meanIndex;
thirdPoint = ((greyDepthmeanIndex)/2)+meanIndex;
```

The second algorithm proposed is a maximum-seeking algorithm. This algorithm seeks to place the midpoint of the colour spectrum at the point in the histogram with the most values assigned to it. This point corresponds to the mode average of the histogram. Pseudocode for the maximum-seeking algorithm follows.

```
//find the index of the maximum histogram

maximum= 0;

for (i = 0; i < greyDepth; I++)
{
  if (histogram[i] > maximum;
 {
    maximum = histogram[i];
    maxIndex = i;
  }
}
```

```
//now set the 3 control points for the colour spectrum

firstPoint = maxIndex/2;
secondPoint = maxIndex;
thirdPoint = ((greyDepthmaxIndex)/2)+maxIndex;
```

The third algorithm used attempted to evenly distribute colour depths based on area under the difference area histogram. The total amount of pixels in the image is calculated from the image dimensions. The total is then divided by the amount of colour partitions desired. The algorithm moves through the histogram, keeping a tally of how many pixels have been found in the histogram. Once the pixel tally exceeds a multiple of the divided total, a breakpoint is set. Here's the pseudocode for this algorithm:

```
areaSeen = 0;

for (i = 0; i <grayDepth; i++)
 {
  areaSeen = areaSeen+ histogram[i];

  //now check area indices
  for (j=0; j<numBreakpoints;j++)
  {
     if((breakpoint[j].isNotSet) \&\& (areaSeen> dividedTotal*j))
     {
        breakpoint[j] = i;
     }
  }
}
```

To test the algorithms, mammogram images from the MIAS database were used. Although sequential images weren't available, differencing two mammograms produced a difference image of sufficient complexity to serve as an evaluation image for these false-colour algorithms. Several mammogram pairings were tested, and results were comparable over all image pairings tested. For demonstration purposes, a left and right mammogram from the same person were differenced. The two mammograms were similar, but because they had been aligned slightly differently on the x-ray machine there were two crescent regions of significant difference where one breast was higher than the other. Another area of large difference occurred where labels appeared. Several fine features were also visible in the breast overlap region which could be used for evaluating fine differences between the images.

Applying the RGB colour spectrum to the relative difference image produced acceptable results. The fine structure inside the common breast region is visible, and the broad positive and negative swaths showed up as strong blue and red artifacts. The fine details in the middle of the common breast region also show up. However, because of the small colour difference, it's hard to tell which of the fine details are positive and which are negative. Applying this colour scheme to an absolute difference image results in an image largely in reds and greens. The only blue visible occurs in the label areas in the upper right. Both images also suffer from some insensitivity: only the brighter differences really come out, and a lot of potentially important "wispy" structure is left hard to distinguish by eye.

The basic HSI colour spectrum highlights contrast a little better than RGB. This is especially apparent around the labels in the upper right hand corner. In RGB, the labels had a smooth transition from the background to the foreground, whereas in the HSI spectrum it is clear from the outline in green or blue that there is a discontinuity in greylevel. The finer structures in the middle of the common breast region are also highlighted clearly, and it is immediately apparent which details are positive and which are negative. The absolute difference image also shows better contrast from the RGB spectra. The crescent of nonoverlapping breast tissues are clearer, and the structures are more apparent to the eye. The largest weakness in the HSI spectrum is that both the low end and high end show up as red. This leaves it up to the interpreter to decide whether a red object is either very similar or very dissimilar through context.

The extended colour spaces produced starker contrasts. Applying blackRGBwhite to the difference image increased contrast between the structures in the difference image. Although there is only a smattering of black or white in the image, up in the labels, extending the colour space squishes the middle colours into the center of the histogram where most of the important details in the image are. The fine difference structures in the middle of the common breast area leap out, as does the wispy detail in the nonoverlapping crescent. The difference is even more dramatic in the absolute difference image. The background becomes black, and the human visual system treats black as a background colour. The fine structures inside the common breast region become very obvious. The altered sequence RGB spectrum, whiteredblackgreenblue, produces the same sort of obvious background for difference images. The fine difference structures appear, and it is easy to tell which is a positive and which is an negative artifact.

The extend hue rotation algorithm also heightened contrast over the standard huerotation algorithm. As with the expanded RGB algorithm, much of this gain resulted from adding more colours to the spectrum, squishing down the colour range into the area of the histogram where the main detail is. Since the extended HSI scheme started with black, it also provides the same black background on absolute difference images.

Because most difference histograms tend to cluster strongly around the zero-difference point, the maximum histogram point and the mean of the histogram tended to be very close together. For all images tested, the mean-seeking algorithm and the maximum-seeking algorithm had similar behaviours. Both the mean-seeking and maximum-seeking algorithms worked very well for relative difference images, since for relative difference images the zero-difference point lies near the middle of the histogram. Both these algorithms placed a strong background colour very close to the zero-difference point in the histogram, which in turn left the colourized image with a clear background colour. Absolute difference images, though, have their zero point at the far left of the histogram. Both mean and maximum seeking algorithms therefore picked their background at the far left of the histogram, which resulted in half the colour space in the spectrum being assigned to only 1 or 2 grey-levels. This skew resulted in much poorer contrast for colourized absolute difference images.

The histogram area matching algorithm performed very well on all test images. Relative difference images had details around the zero-difference point highlighted very strongly while still preserving details in the extremes of the histogram. In the white-red-black-green-blue progression, for example, the frilly detail in the upper half of the non-overlapping crescent is preserved in the green/blue contrast. In the lower half the same frilly detail is preserved inside the red/white contrast. The gray level gradients around the labels are also preserved. The same sort of sharp contrast occurs in the HSI colour space as well. This algorithm produced good results on absolute difference images as well. In both RGB and HSI colour spaces the fine detail inside the common breast area comes out, while the contrast differences around the labels, at the extreme high end of the histogram, are preserved.

False colour images can enhance contrast and bring out detail in gray level images by exploiting colour perception in the human visual system. Static, predefined colour lookup tables are commonly used to generate false colour images. In this paper, several methods of extending false colour generation were proposed and evaluated. An algorithm which sought the mean point in the histogram, an algorithm which sought the maximum point in the histogram, and an algorithm which tried to match colour density with histogram area were proposed. These algorithms were tested on difference images, and their performance in bringing out detail was evaluated.