Exercise: Image and Video Processing

Solution Sheet 1 – Image Similarity

```
void RGB2YUV (double r, double g, double b, double &y, double &Cb, double &Cr)
{
     const double CR = 0.2989;
     const double CG = 0.5866;
     const double CB = 0.1145;

     y  = CR * r + CG * g + CB * b;
     Cb = (b-y) / (2.0 - 2 * CB) + 127.5;
     Cr = (r-y) / (2.0 - 2 * CR) + 127.5;
     return;
}
void YUV2RGB (double y, double Cb, double Cr, double &r, double &g, double &b)
{
     const double CR = 0.2989;
     const double CG = 0.5866;
     const double CB = 0.1145;

     Cb -= 127.5;
     Cr -= 127.5;
     r = Cr * (2.0 - 2 * CR) + y;
     b = Cb * (2.0 - 2 * CB) + y;
     g = (y - CB * b - CR * r) / CG;

     if (r<0) r=0; if (r>255) r=255;
     if (g<0) g=0; if (g>255) g=255;
     if (b<0) b=0; if (b>255) b=255;
     return;
   }
```

```
// sum of absolute difference
void SAD (Image &img1, Image &img2, double &diff)
{
      diff=0;
      for (int y=0; y<height; y++) {
            for (int x=0; x<width; x++) {
                  diff += fabs(img1[0][y][x]-img2[0][y][x]);
            } // for
      } // for
      double cnt = width*height*1;
      diff/=cnt;
      return;
}
```

```
// histogram difference
void HD (Image &img1, Image &img2, double &diff)
{
      // calculate two histograms
      vector<double> hist1, hist2;
      hist1.resize(256);
      hist2.resize(256);
      int i=0;
      for (i=0; i<256; i++) {
            hist1[i]=0;
            hist2[i]=0;
      }
      for (int y=0; y<height; y++) {
            for (int x=0; x<width; x++) {
                  hist1[img1[0][y][x]]++;
                  hist2[img2[0][y][x]]++;
            } // for
      } // for

      // scale histogram
      double factor = 1.0 / (width*height);
      for (i=0; i<256; i++) {
            hist1[i]*=factor;
            hist2[i]*=factor;
      }

      // calculate histogram difference
      diff=0;
      for (i=0; i<hist1.size(); i++) {
            diff+= fabs(hist1[i]-hist2[i]);
      }
      return;
}


// standard deviation
void SD (Image &img1, Image &img2, double &diff)
{
      // average grayscale value
      int avg1=0;
      int avg2=0;
      for (int y=0; y<height; y++) {
            for (int x=0; x<width; x++) {
                  avg1 += img1[0][y][x];
                  avg2 += img2[0][y][x];
            } // for
      } // for
      int factor = width*height;
      avg1 /= factor;
      avg2 /= factor;

      // calculate standard deviation
      double sigma1=0;
      double sigma2=0;
      for (int y=0; y<height; y++) {
            for (int x=0; x<width; x++) {
                  sigma1 += (avg1 - img1[0][y][x]) * (avg1 - img1[0][y][x]);
                  sigma2 += (avg2 - img2[0][y][x]) * (avg2 - img2[0][y][x]);
            } // for
      } // for
      sigma1 /= factor;
      sigma2 /= factor;
      sigma1 = sqrt (sigma1);
      sigma2 = sqrt (sigma2);
      diff =fabs (sigma1 - sigma2);
      return;
}
```