Janne Koljonen
Vaasa yliopisto
1.3.2012
AUTO1030 Signaalien käsittely
Harjoitus 4, tehtävä 5–6

Indeksoidut värit pistetulolla

Tehdään ImageJ-plugin ja siihen metodi, joka laskee pistetulon. Tehdään toinen metodi, joka edellisen metodin avulla laskee annetun RGB-värin pistetulon taulukkona annettujen RGB-värien (väripaletin) kanssa ja palauttaa taulukosta värin, jonka kanssa pistetulo oli suurin. Lopuksi tehdään viimeistellään plugin sellaiseksi, joka muuttaa annetun kuvan indeksoiduksi värikuvaksi seuraavasti:

1. Lue kuvasta RGB-arvo.
2. Laske sen ja väripaletin värien väliset pistetulot.
3. Valitse väripaletista RGB-väri, jonka kanssa pistetulo oli suurin.
4. Korvaa kuvan pikseli kohdasta 3 saadulla väripaletin värillä.
5. Toista kohdat 1–4 kaikille pikseleille.

Huom! Nyt ei siis koodata kuvaa indeksoiduksi kuvaksi, jossa pikselissä olisi viittaus taulukossa (väripaletissa) olevaan väriin, vaan esitetään kuva käyttäen väripaletin värejä.

Tuloksena seuraava koodi:

```
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.filter.*;


// Värikanavien vaihto.
public class IndexedRGB_ implements PlugInFilter {
   ImagePlus imp;

   public int setup(String arg, ImagePlus imp) {
      this.imp = imp;
      return DOES_RGB;
   }

   public void run(ImageProcessor ip) {
      // Kysytään kuvaoliolta kuvan leveyttä ja korkeutta
      int W=ip.getWidth();
      int H=ip.getHeight();

      // Apumuuttujat
      int[] RGB=new int[3];

      int[][] paletti={{0,0,0}, {255, 0, 0}, {0, 255, 0}, {0, 0, 255}, {255, 255, 255},
{0, 127, 255}, {0, 255, 127}, {127, 0, 255}, {127, 255, 0}, {255, 0, 127}, {255, 127,
0}};

      // Käydään jokainen pikseli läpi kahdella sisäkkäisellä for-silmukalla
      for(int y=0; y<H; y++)
      {
         for(int x=0; x<W; x++)
         {
            // Luetaan kuvan ip pikseli (x, y) apumuuttujaan RGB
            ip.getPixel(x,y,RGB);
            // Muutetaan palettiväriksi
            RGB=indeksiVari(RGB, paletti);
            // Kirjoitetaan pikseli RGB takaisin kuvaan ip
            ip.putPixel(x,y, RGB);
         }
      }
```

```
      // Päivitetään kuva näytölle
      imp.updateAndDraw();
   }

   // Metodi vertaa annettua RGB-arvoa ja RGB-palettia pistetulolla.
   // Palauttaa värin, jolla pistetulo on suurin
   // paletti[väri][RGB-arvot]
   private int[] indeksiVari(int[] RGB, int[][] paletti)
   {
      double max_pistetulo=-Integer.MAX_VALUE;
      int[] max_RGB={0, 0, 0};
      // Käydään läpi väripaletti
      for (int i=0; i<paletti.length; i++)
      {
         int[] tempRGB={paletti[i][0], paletti[i][1], paletti[i][2]};
         double pistetulo=pistetulo(RGB, tempRGB);
         if(pistetulo>max_pistetulo)
         {
            max_pistetulo=pistetulo;
            max_RGB=tempRGB;
         }
      }
      return max_RGB;
   }

   // Pistetulo normalisoiduille RGB-väreille
   private double pistetulo(int[] RGB1, int[] RGB2)
   {
      double[] rgb1=normeeraa(RGB1);
      double[] rgb2=normeeraa(RGB2);

      // Pistetulo. Oletetaan, että RGB:n pituus on kolme
      return rgb1[0]*rgb2[0]+rgb1[1]*rgb2[1]+rgb1[2]*rgb2[2];
   }

   // Normeeraus RGB-väreille
   private double[] normeeraa(int[] RGB)
   {
      double pituus=Math.sqrt(RGB[0]^2+RGB[1]^2+RGB[2]^2);
      double[] rgb=new double[3];
      if (pituus != 0)
      {
         rgb[0]=RGB[0]/pituus;
         rgb[1]=RGB[1]/pituus;
         rgb[2]=RGB[2]/pituus;
      }
      return rgb;
   }

}
```