Übung zur Vorlesung Video-Inhaltsanalyse

Blatt 2 – Analyse der Kamerabewegung

Lösung: Aufgabe 1 – Klasse CameraModel

Entwerfen Sie eine Klasse *CameraModel*, die folgende Funktionen zur Verfügung stellt. Die ersten beiden Funktionen transformieren die Position (*srcX/srcY*) eines Pixels in eine neue Position (*destX/destY*).

- **`void transformPointCylindric (double srcX, double srcY, double &destX, double &destY);`**
- **`void transformPointEightParameter (double srcX, double srcY, double &destX, double &destY);`**
- **`void transformCylindric (Image &src, Image &dest);`**
- **`void transformEightParameter (Image &src, Image &dest);`**

```
void transformPointCylindric (double srcX, double srcY, int &destX, int &destY)
{
   // normalize coordinates
   srcX = (srcX -width/2)  / width;
   srcY = (srcY -height/2) / height;

   // rotation
   destX =(int) (width/2.0+width * (atan (srcX / focalLength)));

   // height
   destY =(int) (height/2.0+height*(srcY / sqrt (srcX*srcX + focalLength*focalLength)));
   return;
}


void transformPointEightParameter (double srcX, double srcY, int &destX, int &destY)
{
   destX = (int) ((srcX*a11 + srcY*a12 + tx)/(srcX*b1 + srcY*b2 + 1.0));
   destY = (int) ((srcX*a21 + srcY*a22 + ty)/(srcX*b1 + srcY*b2 + 1.0));
   return;
}
void transformCylindric (Image &src, Image &dest)
{
   width=src.width;
   height=src.height;
   dest.resize  (width, height, src.band);
   dest.clear(0);

   int destX, destY;
   for (int b=0; b<dest.band; b++) {
      for (int h=0; h<dest.height; h++) {
            for (int w=0; w<dest.width; w++) {
                  transformPointCylindric (w, h, destX, destY);
                  if (destX>=0 && destX<dest.width && destY>=0 && destY<dest.height) {
                        dest.data[b][destY][destX] = src.data[b][h][w];
                  } // if
            }
      }
   }
   return;
}
```

```
void transformEightParameter (Image &src, Image &dest)
{
  width=src.width;
  height=src.height;
  dest.resize  (width, height, src.band);
  dest.clear(0);

  int destX, destY;
  for (int b=0; b<dest.band; b++) {
       for (int h=0; h<dest.height; h++) {
             for (int w=0; w<dest.width; w++) {
                   transformPointEightParameter (w, h, destX, destY);
                   if (destX>=0 && destX<dest.width && destY>=0 && destY<dest.height) {
                         dest.data[b][destY][destX] = src.data[b][h][w];
                   } // if
             }
       }
  }
  return;
}
```

Lösung: Aufgabe 2 – Klasse CameraModel

Erweitern Sie die Klasse CameraModel, um die Verschiebung zweier Bilder zu berechnen (*tx* und *ty* beim 8-Parameter Modell). Entwickeln Sie dazu folgende Funktion:

- **`void getTranslation (Image &img1, Image &img2, int &tx, int &ty);`**

```
void getTranslation (Image &img1, Image &img2, int &tx, int &ty)
{
  int minDiff = 100000000;
  for (int y=-height+1; y<height+1; y++) {
       for (int x=-width+1; x<width-1; x++) {

             int diff=0;
             int cnt=0;

             for (int h=0; h<height-1; h++) {
                   for (int w=0; w< width+1; w++) {

                     int r1=img1.data[0][h][w];
                     int g1=img1.data[1][h][w];
                     int b1=img1.data[2][h][w];
                     if (r1==255 && g1==255 && b1==128) continue; // undefined pixel

                     int r2=img2.data[0][y+h][x+w];
                     int g2=img2.data[1][y+h][x+w];
                     int b2=img2.data[2][y+h][x+w];
                     if (r2==255 && g2==255 && b2==128) continue; // undefined pixel

                     diff += abs (img1.data[0][h][w] - img2.data[0][py][px]) +
                             abs (img1.data[1][h][w] - img2.data[1][py][px]) +
                             abs (img1.data[2][h][w] - img2.data[2][py][px]);
                     cnt++;
                   } // for
             } // for
             diff /= cnt;
             if (minDiff>diff) {
                   tx=x;
                   ty=y;
                   minDiff=diff;
             } // if
       } // for
  } // for
  return;
}
```