Real-Time Panorama Stitching using a Single PTZ-Camera without Image Feature Matching

Rikard Lindahl, Hampus Linse

When monitoring a large area very often a camera has limits in how much it can see. With our algorithms we can rotate a pan-tilt camera in high speeds and put images from the sequence together into a larger image. A quick panorama-maker.

Often when surveying an area, like a car park or airport terminal, one would like to get an image covering as large portion of the area as possible. Preferably while keeping high quality. Our algorithm combines high quality images and relevant image data, like the orientation of the camera, to produce a half sphere panorama view. In practice this means we have enhanced the field of view for a camera up to 180x180 degrees, although at the loss of frame rate and a few other limiting factors. A full half sphere panorama can be captured in 1.6 seconds. A resulting image from the beautiful Lundagård in Lund taken in 4 seconds is presented in Figure 1. We have used 65 images captured in different angles to produce this image.

Figure 1: The scenic Lundagård.

With this technology one can create an overview of the scene for navigating the camera, that is a small panorama image the operator can click in and the camera will move to this direction. One can also enhance the field of view of the camera in different ways. If the camera starts spinning around the direction it is looking at we can cover more directions and therefore "zoom out" more than the camera is usually able to do. Other panorama representations can also be used.

Most panorama stitching methods use feature point matching. This means they find points in images that look the same and try to match the images so these points fit. These methods are quite computationally heavy. We on the other hand only use data from the camera, like orientation and moving speed, to calculate where the image should be. This makes the calculations faster and with good precision of the data we get comparatively good results.

Distortions that we had to compensate for was lens distortion, intensity differences and distortions from moving the camera at high speeds (so called rolling shutter effects). Even with all these corrections there were visible seams between the images and some pixel mismatches. Therefore we blended the images seamlessly using a technique called "multi-band blending". Although we didn't manage to do this in real-time we are confident it can be done if our algorithms are optimized and implemented in faster programming language than the one we used.