Exercise on image stiching using IMU

xiahaa@space.dtu.dk

March 31, 2019

In this exercise, you will work on using IMU of image stiching. IMU usually refers to Inertial Measurement Unit.

1 IMU

If you search with Google or Alibaba, then probably you will find something with a similar name but with a prefix like 3-axis, 6-axis, and 9-axis which corresponds to the sensors integrated inside the IMU. For example, if one IMU only contains 3-axis gyroscope (for measuring angular velocity), then it will be called as 3-axis IMU. If besides gyroscope, it also includes a 3-axis accelerometer (for measuring acceleration) and will be called as 6-axis IMU.

High-end IMU is very accurate and expensive, which is often used in aeroplanes, fighters, missiles. Low-end IMU is very cheap and widely used in various products, like mobile-phones, action cameras, AR, etc.

In this exercise, we will only use the data provided by low-end IMU. The data is obtained from the ESE650 given at University of Pennsylvania[1]. Sensor fusion is needed in order to get accurate accurate information. Code has been provided to you for this part. The filter used here is the so-called complementary filter on Special Orthogonal Group ($\mathbb{SO}(3)$). An example of the sensor fusion result is shown in Figure 1. VICON is

[1] https://upenn.box.com/v/ese650Proj2-train

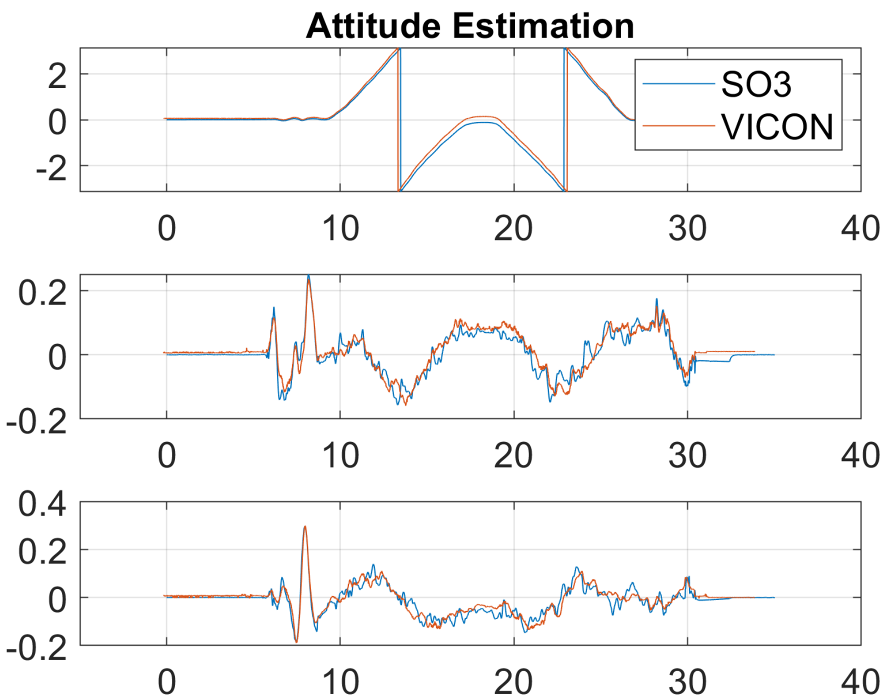

Figure 1: Example of attitude estimation result.

a motion capturing system which can be used as the ground truth. As can be seen, the fusion algorithm is fairly good and the errors are quite small.

2 IMU aided image stiching

Since low-end IMU can only give us attitude information, the image stiching will be done using the cylindrical projection. Cylindrical projection is nothing but project each pixel into a cylindrical coordinate system. In order to accomplish this, you need to do as follows:

1. estimate the camera focus length f. A rough estimation of f can be done as

$$f = \frac{0.5width}{tan(\frac{FOV}{2})}$$

   where $width$ means the image width, FOV is the Field-Of-View (you can use $\frac{\pi}{3}$ here).

2. The cylindrical projection is shown in Figure 2. So basically, you need to transform every pixel of the image to its corresponding point on cylindrical surface. This has to be done in several steps:

   - move origin to principle point, so $(u, v) \rightarrow (u - c_u, v - c_v)$ (so point on the left (right) will have a negative (positive) x coordinate).
   - point in camera frame to point in IMU frame $\mathbf{p}_{imu} = \mathbf{R}_{cam}^{imu}\mathbf{p}_{cam}$, here $\mathbf{R}_{cam}^{imu} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$.
   - point in Cartesian frame to Cylindrical frame

$$\begin{aligned} r &= \sqrt{(}x^2 + z^2) \\ x &= \frac{x}{r} \\ y &= \frac{y}{z} \\ z &= \frac{z}{r} \end{aligned}$$

   - to point in initial global cylindrical frame $\mathbf{p}_{glb} = \mathbf{R}_{imu}^{glb}\mathbf{p}_{imu}$, here $\mathbf{R}_{imu}^{glb}$ is from sensor fusion algorithm.
   - to θ, h by $\theta = -atan2(z, x)$ and $h = -y$ (negative because the top left starts with $(1, 1)$ in the image).
   - add cu, cv. cu, cv is the center of the stiched image. For example, you can use $\frac{w}{2}, \frac{h}{2}$ as the center.

   Tips that can make this process faster: the first three steps are constant, so they only need to be done once.

   This may be complex for you. So if you have problem, just ask.

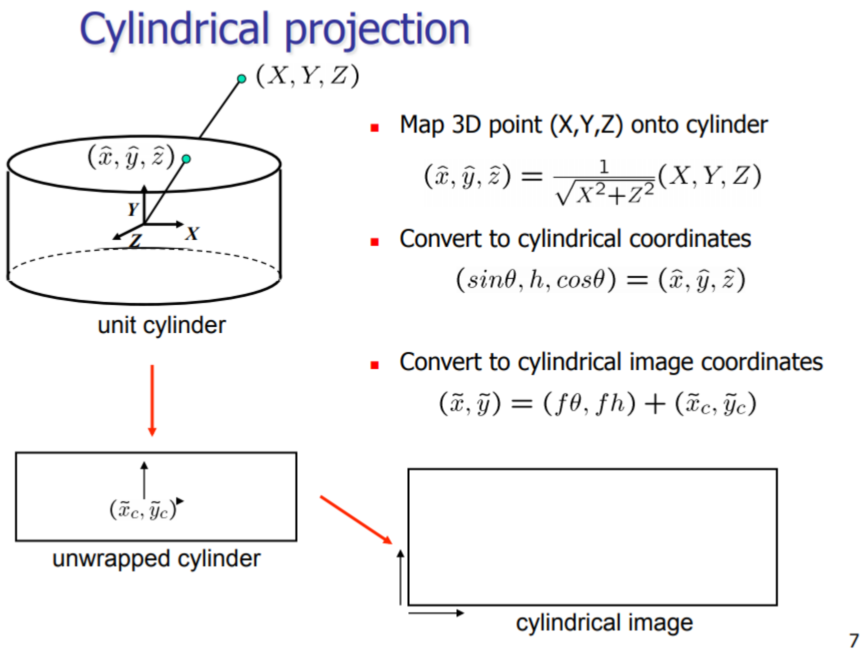

Figure 2: Cylindrical projection, figure from https://cs.gmu.edu/~kosecka/cs482/lect-panoramas.pdf.