阅读报告

By Weiheng Xia

1. 论文所解决的问题

如何用spline(样条)插值方法生成continuous-time trajectory，而这个cubic spline的方法能够通过控制spline方程的参数如下图1公式所示，来让生成的trajectory平滑贴近真实Rolling-shutter相机的运动轨迹模型，如图2所示。

$$\tilde{\mathbf{B}}(u) = \mathbf{C}\begin{bmatrix}1\\u\\u^2\\u^3\end{bmatrix}, \quad \dot{\tilde{\mathbf{B}}}(u) = \frac{1}{\Delta t}\mathbf{C}\begin{bmatrix}0\\1\\2u\\3u^2\end{bmatrix}, \quad \ddot{\tilde{\mathbf{B}}}(u) = \frac{1}{\Delta t^2}\mathbf{C}\begin{bmatrix}0\\0\\2\\6u\end{bmatrix}, \quad \mathbf{C} = \frac{1}{6}\begin{bmatrix}6 & 0 & 0 & 0\\5 & 3 & -3 & 1\\1 & 3 & 3 & -2\\0 & 0 & 0 & 1\end{bmatrix}$$

The pose in the spline trajectory can now be defined as:

$$\mathbf{T}_{w,s}(u) = \mathbf{T}_{w,i-1}\prod_{j=1}^{3}\exp\left(\tilde{\mathbf{B}}(u)_j\,\Omega_{i+j}\right), \tag{4}$$

图1. Cumulative Cubic B-Splines **[1]**

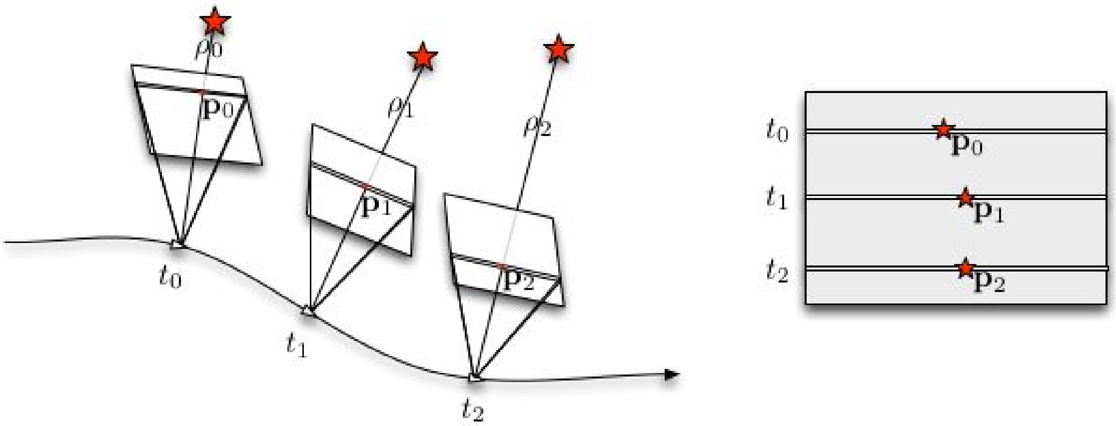

图2. Rolling-shutter Camera Trajectory **[1]**

2. 论文的基本思路

首先，我们应用图1公式中的Cubic B-Splines可以得到轨迹上的位姿Pose:

$$\mathbf{T}_{w,s}(u) = \mathbf{T}_{w,i-1} \prod_{j=1}^{3} \exp\left(\tilde{\mathbf{B}}(u)_j \Omega_{i+j}\right)$$

其次，我们可以通过 *i-1* 时刻的位姿得到 *i* 时刻的位姿，并很容易求它的一阶导数和二阶导数，即速度和加速度，如下面的公式所示:

$$\mathbf{T}_{w,s}(u) = \mathbf{T}_{w,i-1}\left(\dot{\mathbf{A}}_0\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_0\dot{\mathbf{A}}_1\mathbf{A}_2 + \mathbf{A}_0\mathbf{A}_1\dot{\mathbf{A}}_2\right)$$

$$\ddot{\mathbf{T}}_{w,s}(u) = \mathbf{T}_{w,i-1}\begin{pmatrix} \ddot{\mathbf{A}}_0\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_0\ddot{\mathbf{A}}_1\mathbf{A}_2 + \mathbf{A}_0\mathbf{A}_1\ddot{\mathbf{A}}_2 + \\ 2\left(\dot{\mathbf{A}}_0\mathbf{A}_1\mathbf{A}_2 + \dot{\mathbf{A}}_0\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_0\mathbf{A}_1\dot{\mathbf{A}}_2\right) \end{pmatrix}$$

$$\mathbf{A}_j = \exp\left(\Omega_{i+j}\tilde{\mathbf{B}}(u)_j\right), \quad \dot{\mathbf{A}}_j = \mathbf{A}_j\Omega_{i+j}\dot{\mathbf{B}}(u)_j$$
$$\ddot{\mathbf{A}}_j = \mathbf{A}_j\Omega_{i+j}\hat{\mathbf{B}}(u)_j + \mathbf{A}_j\Omega_{i+j}\tilde{\mathbf{B}}(u)_j$$

最后，我们可以将这个样条曲线Cubid Spline应用到Visual Odometry上。相邻两帧a, b之间的位姿可以表示为Pb：

$$\mathbf{p}_b = \mathcal{W}(\mathbf{p}_a; \mathbf{T}_{b,a}, \rho) = \pi\left(\left[\mathbf{K}_b \,|\, \mathbf{0}\right] \mathbf{T}_{b,a} \left[\mathbf{K}_a^{-1} \begin{bmatrix} \mathbf{p}_a \\ 1 \end{bmatrix}; \rho\right]\right),$$

通过上面得到的Cubic B-Spline轨迹公式，我们很容易得到Accelerator和Gyro的轨迹模型，如下所示：

$$
\begin{aligned}
\operatorname{Gyro}(u) &= \mathbf{R}_{w,s}^{\top}(u) \cdot \dot{\mathbf{R}}_{w,s}(u) + \text{ bias} \\
\operatorname{Accel}(u) &= \mathbf{R}_{w,s}^{\top}(u) \cdot \left(\mathbf{s}_w(u) + g_w\right) + \text{ bias}
\end{aligned}
$$

得到这个轨迹后，我们通过最小化误差函数E来得到Spline的轨迹参数:

$$
\begin{aligned}
E(\theta) = & \sum_{\hat{\mathbf{p}}_m} \left(\hat{\mathbf{p}}_m - \mathcal{W}\left(\mathbf{p}_r; \mathbf{T}_{c,s}\left(u_m\right)^{-1} \mathbf{T}_{w,s}\left(u_r\right) \mathbf{T}_{s,c}, \rho\right)\right)_{\Sigma_p}^2 + \\
& \sum_{\hat{\omega}_m} \left(\hat{\omega}_m - \operatorname{Gyro}\left(u_m\right)\right)_{\Sigma_\omega}^2 + \sum_{\hat{\mathbf{a}}_m} \left(\hat{\mathbf{a}}_m - \operatorname{Accel}\left(u_m\right)\right)_{\Sigma_{\mathbf{a}}}^2
\end{aligned}
$$

3. 论文的应用范围

这篇论文的Cubic B-Spline Parametrization的方法可以应用在多传感器的vSLAM以及Calibration上，尤其适合用于对时间连续性要求高的Rolling-shutter(卷帘快门)相机轨迹生成。他能够通过Cubic B-Spline来拟合卷帘快门相机的轨迹，并用最小二乘法来异步的内外参。

参考文献:

[1] Lovegrove, Steven & Patron-Perez, Alonso & Sibley, Gabe. (2013). Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. BMVC 2013 - Electronic Proceedings of the British Machine Vision Conference 2013. 93.1-93.11. 10.5244/C.27.93.