

基于姿态传感器紧耦合的视频稳像算法

陈寿峰¹, 刘宇², 唐武军¹, 张逸飞¹, 王乐豪¹

(1. 中国铁塔股份有限公司无锡分公司, 江苏 无锡 214111; 2. 中国铁塔股份有限公司南京科创中心, 南京 211316)

摘要: 由于特征点或光流匹配与图像的纹理丰富程度有关,使用单独的视觉方法进行视频稳像无法适用于所有场景。提出一种基于传感器紧耦合的视频稳像算法,通过自适应调节权重,使得图像间最优单应矩阵在弱纹理区域更加依赖于姿态传感器,在纹理丰富区域更加依赖于特征点匹配。在得到最优的图像变换之后,使用鲁棒弹性映射方法进一步配准连续的图像帧。实验表明,提出的视频稳像算法有更佳的效果和鲁棒性。

关键词: 姿态传感器; 鲁棒弹性映射; 视频稳像

中图分类号: TP751.1 **文献标志码:** A **文章编号:** 1671-1807(2025)13-0057-07

通过将相机安装在楼顶、铁塔等高处,工厂、学校、农田和机场^[1]等区域可以获得更大视野的监控视频。在实际生产环境中,由于铁塔普遍高达三五十米,加上铁塔本身的结构问题^[2],极易受到风雪等天气因素的影响,造成视频的抖动,进而影响视频质量和用户观感。基于图像的视频稳像方法是解决视频抖动的重要手段^[3-6]。

基于图像的视频稳像方法或者通过提高特征点匹配的精度提升稳像质量^[7-8],或者通过将图像网格化,并添加内容保持约束、特征点对齐约束、光度一致性约束和平滑约束提升稳像算法的稳定性^[9-14]。或者采用深度学习方法,通过构建网络结构以非监督学习方式学习光流或者单应矩阵,使得稳像后的视频帧间像素差最小^[15-16]。基于图像的视频稳像结果严重依赖于采用的图像运动模型,如全局运动模型或者局部运动模型。而运动模型的计算需要使用匹配的特征点或者图像间的光流。在弱纹理场景下(如天空、建筑墙面等),特征点检测匹配和光流计算都会出现较大的误差,进而影响后续的视频稳像效果。

姿态传感器可以不依赖于环境监测目标物体的运动^[17]。借助于姿态传感器和相机的互补性,在SLAM(simultaneous localization and mapping,同时定位和地图构建)领域,通过耦合姿态传感器和图像,取得了巨大的成功^[18]。类似的,借助姿态节点,

可以提高视频稳像算法的稳定性。因此本文提出基于姿态传感器紧耦合的视频稳像算法。通过姿态传感器角速度的积分可以得到两帧图像间的旋转变换。之后通过降低匹配特征点的重投影误差,以及降低最优变换与IMU(inertial measurement unit,惯性测量单元)积分计算的旋转之间的差异,得到最优的图像变换。重投影误差的权重由图像纹理丰富程度动态决定。当纹理细节较丰富时,特征点匹配的置信度较高,重投影误差的权重更大。面对弱纹理场景时,特征点匹配的置信度较低,最优图像变换更加依赖于姿态节点计算的旋转变换。通过这种方法,使得本文提出的算法可以应对更多的场景。之后,通过将鲁棒弹性映射应用于使用单应矩阵对齐后的图像对,可以更好地对齐图像,生成的稳像图像也更佳。

本文的主要贡献为:将图像和姿态传感器紧耦合,通过动态调节目标函数中重投影误差项的权重,使得视频稳像算法在弱纹理场景时可以依赖姿态传感器得到更鲁棒的结果;通过单应矩阵映射图像之后,使用鲁棒弹性映射进一步降低帧间像素差。

1 视频稳像方法

在视频监控领域,拍摄相机需要保持静止,且一般情况下,拍摄场景中前景目标较少,因此视频稳像方法可以转换为图像对齐问题。图像间的单应矩阵变换是图像对齐的重要方法。

收稿日期: 2025-01-02

作者简介: 陈寿峰(1977—),男,江苏无锡人,硕士,高级工程师,研究方向为云计算、大数据和物联网;刘宇(1989—),男,江苏徐州人,硕士,架构师,研究方向为云计算、分布式存储、云桌面;唐武军(1980—),男,安徽绩溪人,硕士,高级工程师,研究方向为网络安全;张逸飞(1994—),男,江苏无锡人,中级工程师,研究方向为图像处理 and 边缘计算;王乐豪(1996—),男,江苏连云港人,硕士,中级工程师,研究方向为视频监控、网络安全。

1.1 图像单应矩阵变换和旋转矩阵

如图 1 所示, 假定相机无畸变(畸变的图像可以通过标定得到畸变参数, 然后变形得到无畸变图像), $t-1$ 时刻的图像为 I_1 , 对应的相机的光心为 O_1 , t 时刻的图像为 I_2 , 对应的相机的光心为 O_2 , 相机的内参矩阵为 K , $t-1$ 时刻到 t 时刻相机的旋转为 R , 平移为 t , 图像 I_1 中坐标 u_1 对应的场景 3D 点坐标 p 所在的平面为 (n, d) , 即 $n^T p + d = 0$, 那么 $t-1$ 时刻图像 I_1 中的像素 u_1 在 t 时刻图像 I_2 中的对应像素 u_2 为

$$\tilde{u}_2 = K \left(R - \frac{Tn^T}{d} \right) K^{-1} \tilde{u}_1 \quad (1)$$

式中: \tilde{u}_2 和 \tilde{u}_1 为 u_1 和 u_2 的齐次坐标。

对于视频稳像场景, 不同时刻相机之间的平移很小, 同时相机距离拍摄目标距离较远, 因此 $\frac{Tn^T}{d} \approx 0$, 式可变为

$$\tilde{u}_2 = H \tilde{u}_1 = K R K^{-1} \tilde{u}_1 \quad (2)$$

相机的内参矩阵可以通过标定^[19]提前获得, 因此为了计算最优的单应矩阵 H , 只需要估计两个相机间的最优旋转矩阵。

由于 3×3 的旋转矩阵是正交矩阵, 矩阵的 9 个元素需要满足一定的条件才能成为旋转矩阵, 直接优化旋转矩阵会增加计算复杂度。任何旋转矩阵都可以表达为绕某个轴旋转一定的角度, 即旋转矩阵可以表达为旋转向量 $\mathcal{O} = \theta n$, \mathcal{O} 的归一化单位向量 n 表示旋转轴, \mathcal{O} 的模 θ 表示旋转的弧度。旋转向量和旋转矩阵之间的转换关系为

$$R = \exp(\mathcal{O}) = \cos\theta I + (1 - \cos\theta) n n^T + \sin\theta n^\wedge \quad (3)$$

$$\theta = \arccos\left(\frac{\text{tr}R - 1}{2}\right) \quad (4)$$

$$Rn = n \quad (5)$$

式中: $\exp(\mathcal{O})$ 为矩阵 \mathcal{O} 的幂矩阵(以 e 为基); I 为单位矩阵; \mathcal{O}^\wedge 为 \mathcal{O} 的反对称矩阵; $\text{tr}R$ 为 R 的迹。

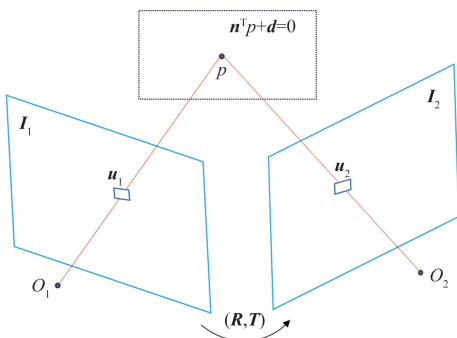


图 1 图像单应矩阵变换

旋转向量 3 个元素之间没有内部约束, 相对于旋转矩阵, 旋转向量的优化更加便捷。

1.2 使用姿态传感器计算旋转矩阵

通过姿态传感器可以获取两帧图像间的旋转矩阵。由于姿态传感器的频率远大于图像帧率, 因此需要对一段时间内的姿态传感器数据进行积分获取两帧之间的旋转。

如图 2 所示, 若两帧相邻图像的获取时间为 t_k^l 和 $t_{k+1}^l, t_k^g, t_{k+1}^g, \dots, t_{j+1}^g$ 时刻的传感器角速度数据为 $\omega_k^g, \omega_{k+1}^g, \dots, \omega_{j+1}^g, t_k^l$ 位于 t_k^g 和 t_{k+1}^g 之间, t_{k+1}^l 位于 t_k^g 和 t_{j+1}^g 之间。 t_k^l 和 t_{k+1}^l 时刻的角速度数据 ω_k^l 和 ω_{k+1}^l 可以通过线性插值计算得到。将 t_k^l 到 t_{k+1}^l 之间的角速度数据进行积分就可以得到两帧图像之间的旋转矩阵 R_k 。即令:

$$t_s = \begin{cases} t_k^l, & s = 0 \\ t_{s+i}^g, & 0 < s \leq j - i \\ t_{k+1}^l, & s = j - i + 1 \end{cases} \quad (6)$$

$$\omega_s = \begin{cases} \omega_k^l, & s = 0 \\ \omega_{s+i}^g, & 0 < s \leq j - i \\ \omega_{k+1}^l, & s = j - i + 1 \end{cases} \quad (7)$$

那么旋转矩阵 R_k 为

$$R_k = \prod_{s=0}^{j-i} \exp[\tilde{\omega}_s (t_{s+1} - t_s)] \quad (8)$$

式中: $\tilde{\omega}_s = \frac{1}{2}(\omega_{s+1} - \eta_{s+1}^g + \omega_s - \eta_s^g)$ 为 t_s 到 t_{s+1} 之间积分所用的角速度, η_s^g 为 t_s 时刻的角速度偏移。一般来说, 通过定时校准, 角速度偏移 η_s^g 可以近似为 0。

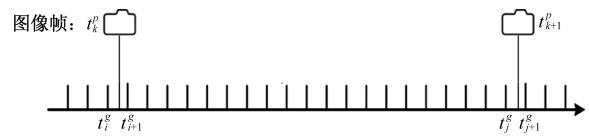


图 2 姿态传感器积分

1.3 构建最优旋转矩阵的代价函数

由于姿态传感器数据存在误差, 因此单纯依靠姿态传感器数据计算图像之间的旋转也会存在问题。通过图像特征点检测、描述和匹配^[20-21], 结合匹配特征点的重投影误差和与姿态传感器计算的旋转矩阵的偏移误差, 可以使用最优化方法计算最优的旋转矩阵。

若 I_1 中的像素 $p_i (i = 1, 2, \dots, n)$ 和 I_2 中的像素 $q_i (i = 1, 2, \dots, n)$ 对应, 通过最小化式中的 E 得到最优的由向量 \mathcal{O} 表示的旋转矩阵。

$$e_{li}(\boldsymbol{\theta}) = f(\mathbf{p}_i, \boldsymbol{\theta}) - \mathbf{q}_i \quad (9)$$

$$\mathbf{E}_1(\boldsymbol{\theta}) = \sum_{i=1}^n \mathbf{e}_{li}^T \mathbf{e}_{li} \quad (10)$$

$$\mathbf{E}_2(\boldsymbol{\theta}) = (\boldsymbol{\theta} - \boldsymbol{\theta}_k)^T (\boldsymbol{\theta} - \boldsymbol{\theta}_k) \quad (11)$$

$$\mathbf{E}(\boldsymbol{\theta}) = \frac{\mu}{n} \mathbf{E}_1 + \mathbf{E}_2 \quad (12)$$

式中: \mathbf{E}_1 为所有匹配特征点的重投影误差; \mathbf{e}_{li} 为第 i 对匹配特征点的重投影误差; \mathbf{E}_2 为待求的最优的旋转向量与姿态传感器计算的旋转向量的偏差; $\boldsymbol{\theta}_k$ 为姿态传感器计算的旋转矩阵 \mathbf{R}_k 对应的旋转向量; $f(\mathbf{p}_i, \boldsymbol{\theta})$ 为将单应矩阵 $\mathbf{K} \exp(\boldsymbol{\theta}) \mathbf{K}^{-1}$ 代入式(2), 得到图像 \mathbf{I}_1 中像素 \mathbf{p}_i 在图像 \mathbf{I}_2 中的对应像素; μ 为由特征点匹配计算单应矩阵的置信度。通过 RANSAC(random sample consensus, 随机抽样一致性算法)使用匹配的特征点计算单应矩阵时, 令总的匹配数为 n_l , 内点数为 n_{inlier} , 那么置信度 μ 为^[22]

$$\mu = \frac{n_{inlier}}{8 + 0.3n_l} \quad (13)$$

从式(13)中可以看出, 当图像中可以检测到的匹配特征点越多, 且其中内点数越多, 置信度越高。显然, 在式表示的代价函数 E 中, μ 越大, 重投影误差的权重越高, 此时, 最优的旋转矩阵更倾向于使重投影误差更小。反之, μ 越小, 最优的旋转矩阵更接近于姿态传感器计算出的旋转矩阵。

1.4 最优化代价函数

通过非线性优化最小化式(12)时, 从初始值 $\boldsymbol{\theta}_k$ 开始, 然后迭代更新旋转向量, 使得目标函数更小。将 $E(\boldsymbol{\theta} + \Delta\boldsymbol{\theta})$ 进行泰勒展开:

$$\mathbf{E}(\boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \approx \mathbf{E}(\boldsymbol{\theta}) + \mathbf{g} \Delta\boldsymbol{\theta} + \frac{1}{2} \Delta\boldsymbol{\theta}^T \mathbf{H} \Delta\boldsymbol{\theta} \quad (14)$$

使用阻尼牛顿方法, 每次的更新量为

$$\Delta\boldsymbol{\theta} = -(\mathbf{H} + \alpha \mathbf{I})^{-1} \mathbf{g} \quad (15)$$

式中: \mathbf{I} 为单位矩阵; \mathbf{H} 为 $\mathbf{E}(\boldsymbol{\theta})$ 关于旋转向量 $\boldsymbol{\theta}$ 的海森矩阵; \mathbf{g} 为 $\mathbf{E}(\boldsymbol{\theta})$ 关于旋转向量 $\boldsymbol{\theta}$ 的梯度向量; α 为阻尼系数。得到 $\Delta\boldsymbol{\theta}$ 之后, 若 $\|\Delta\boldsymbol{\theta}\|$ 小于指定阈值, 则停止优化。不然令 $\boldsymbol{\theta} = \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$, 代入式(14), 得到新的更新量 $\Delta\boldsymbol{\theta}$, 重复这个过程, 直至 $\|\Delta\boldsymbol{\theta}\|$ 小于指定阈值。关于 \mathbf{H} 和 \mathbf{g} 的计算如下:

$$\mathbf{H}_{li}(\boldsymbol{\theta}) \approx \mathbf{J}_{li}^T(\boldsymbol{\theta}) \mathbf{J}_{li}(\boldsymbol{\theta}) \quad (16)$$

$$\mathbf{H}_2(\boldsymbol{\theta}) \approx \mathbf{J}_2^T(\boldsymbol{\theta}) \mathbf{J}_2(\boldsymbol{\theta}) \quad (17)$$

$$\mathbf{H}(\boldsymbol{\theta}) = \frac{\mu}{n} \sum_{i=1}^n \mathbf{H}_{li} + \mathbf{H}_2 \quad (18)$$

$$\mathbf{g}_{li}(\boldsymbol{\theta}) = \mathbf{J}_{li}^T(\boldsymbol{\theta}) e_{li}(\boldsymbol{\theta}) \quad (19)$$

$$\mathbf{g}_2(\boldsymbol{\theta}) = \mathbf{J}_2^T(\boldsymbol{\theta}) (\boldsymbol{\theta} - \boldsymbol{\theta}_k) \quad (20)$$

$$\mathbf{g}(\boldsymbol{\theta}) = \frac{\mu}{n} \sum_{i=1}^n \mathbf{g}_{li} + \mathbf{g}_2 \quad (21)$$

式中: $\mathbf{H}_{li}(\boldsymbol{\theta})$ 为 $\mathbf{e}_{li}^T \mathbf{e}_{li}$ 关于 $\boldsymbol{\theta}$ 的海森矩阵; $\mathbf{H}_2(\boldsymbol{\theta})$ 为 $\mathbf{E}_2(\boldsymbol{\theta})$ 关于 $\boldsymbol{\theta}$ 的海森矩阵; \mathbf{J}_{li} 为 \mathbf{e}_{li} 关于 $\boldsymbol{\theta}$ 的雅克比矩阵; $\mathbf{J}_2(\boldsymbol{\theta})$ 为 $(\boldsymbol{\theta} - \boldsymbol{\theta}_k)$ 关于 $\boldsymbol{\theta}$ 的雅克比矩阵。由于海森矩阵涉及代价函数关于优化变量的二阶导数, 计算复杂度较高, 因此一般采用高斯牛顿方法中的 $\mathbf{J}^T \mathbf{J}$ 近似海森矩阵 \mathbf{H} 。

显然, $\mathbf{J}_2(\boldsymbol{\theta}) = \mathbf{I}$ 。而对于 $\mathbf{J}_{li}(\boldsymbol{\theta})$, 则使用链式法则, 计算 \mathbf{e}_{li} 关于各个参数的导数, 进而获得雅克比矩阵。

$$\frac{\partial \mathbf{e}_{li}}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{e}_{li}}{\partial \mathbf{p}'_i} \frac{\partial \mathbf{p}'_i}{\partial \tilde{\mathbf{p}}'_i} \frac{\partial \tilde{\mathbf{p}}'_i}{\partial \boldsymbol{\theta}} \quad (22)$$

式中: $\tilde{\mathbf{p}}'_i = \mathbf{K} \exp(\boldsymbol{\theta}) \mathbf{K}^{-1} \tilde{\mathbf{p}}_i$, $\tilde{\mathbf{p}}_i$ 为 \mathbf{p}_i 的齐次坐标表示; \mathbf{p}'_i 为 $\tilde{\mathbf{p}}'_i$ 的非齐次坐标。若 $\tilde{\mathbf{p}}'_i = (x, y, z)^T$, 那么 $\frac{\partial \mathbf{e}_{li}}{\partial \mathbf{p}'_i}$ 和 $\frac{\partial \mathbf{p}'_i}{\partial \tilde{\mathbf{p}}'_i}$ 的雅克比矩阵为

$$\frac{\partial \mathbf{e}_{li}}{\partial \mathbf{p}'_i} = \mathbf{I} \quad (23)$$

$$\frac{\partial \mathbf{p}'_i}{\partial \tilde{\mathbf{p}}'_i} = \frac{\partial \begin{bmatrix} x \\ y \\ z \end{bmatrix}}{\partial \begin{bmatrix} x \\ y \\ z \end{bmatrix}} = \begin{bmatrix} 1/z & 0 & -x/z^2 \\ 0 & 1/z & -y/z^2 \end{bmatrix} \quad (24)$$

继续使用链式法则计算 $\tilde{\mathbf{p}}'_i$ 关于 $\boldsymbol{\theta}$ 各个分量的导数, 如 $\tilde{\mathbf{p}}'_i$ 关于 $\boldsymbol{\theta}$ 的第一个分量 $\boldsymbol{\theta}_1$ 的导数为

$$\frac{\partial \tilde{\mathbf{p}}'_i}{\partial \boldsymbol{\theta}_1} = \mathbf{K} \frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}_1} \mathbf{K}^{-1} \tilde{\mathbf{p}}_i \quad (25)$$

$$\frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}_1} = \frac{\partial}{\partial \boldsymbol{\theta}_1} e^{[\boldsymbol{\theta}]_{\times}} = e^{[\boldsymbol{\theta}]_{\times}} \frac{\partial [\boldsymbol{\theta}]_{\times}}{\partial \boldsymbol{\theta}_1} = e^{[\boldsymbol{\theta}]_{\times}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (26)$$

类似的, 可以计算出, $\tilde{\mathbf{p}}'_i$ 关于 $\boldsymbol{\theta}_2$ 和 $\boldsymbol{\theta}_3$ 的导数,

$$\frac{\partial \tilde{\mathbf{p}}'_i}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \tilde{\mathbf{p}}'_i}{\partial \boldsymbol{\theta}_1} & \frac{\partial \tilde{\mathbf{p}}'_i}{\partial \boldsymbol{\theta}_2} & \frac{\partial \tilde{\mathbf{p}}'_i}{\partial \boldsymbol{\theta}_3} \end{bmatrix}。$$

综上, 计算出所有的雅克比矩阵、海森矩阵之后, 可以迭代优化相机间的旋转向量。

旋转矩阵的优化流程如图 3 所示。即将图像初始时刻的旋转矩阵设置为单位矩阵, 并检测当前图像的特征点, 在下一帧图像到来之前, 将收到的姿态传感器角速度数据进行逐步积分。当新的一帧图像到来之后, 检测新一帧图像的特征点, 并与之前图像的特征点进行匹配, 得到匹配的特征点对, 和姿态传感器积分得到的旋转矩阵一起代入式(12)进行优化。



图 3 旋转矩阵优化流程

1.5 弹性映射

联合姿态传感器和图像,可以估计出最优的旋转矩阵。但是由于相机之间存在位移,或者相机的畸变系数计算误差,式(2)无法准确表征图像间像素的映射。参考鲁棒弹性映射^[23],通过对图像进行进一步变形,可以得到误差更低的映射图像。

设两幅图像 I_1 和 I_2 内匹配的特征点为 $p_i = (x_i, y_i)^T, q_i = (u_i, v_i)^T (i = 1, 2, \dots, n)$, I_1 和 I_2 之间的全局单应矩阵为 H , 记 $p'_i = (x'_i, y'_i)^T$ 为 $p_i = (x_i, y_i)^T$ 使用全局单应矩阵 H 映射后在 I_2 中的投影点。 p'_i 处的投影偏差为 $f_i = p'_i - q_i$ 。若图像之间存在视差,那么 $f_i \neq 0$ 。弹性映射方法通过匹配特征点的投影偏差,计算每个网格的偏差。鲁棒弹性映射方法使用薄板样条(thin plate spline, TPS)^[23]插值方法从稀疏的匹配特征点中计算得到的图像之间的映射偏差 $f(x, y)$ 为

$$f(x, y) = \begin{bmatrix} \sum_{i=1}^n w_{1i} \phi(r_i) + a_{11}x + a_{12}y + a_{13} \\ \sum_{i=1}^n w_{2i} \phi(r_i) + a_{21}x + a_{22}y + a_{23} \end{bmatrix} \quad (27)$$

式中: $r_i = \|p - p'_i\|_2$ 为像素 $p(x, y)$ 距离 p'_i 的距离; $\phi(r_i) = r_i^2 \ln r_i^2$ 为径向基函数; $w_{ij} (i = 1, 2, j = 1, 2, 3)$ 和 $a_{ij} (i = 1, 2; j = 1, 2, 3)$ 可以通过求解如下的线性方程计算得到:

$$\begin{pmatrix} \mathbf{K} + 8\lambda\pi\mathbf{I} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{a} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \quad (28)$$

式中: $\mathbf{K} = [\varphi(r_{ij})]_{n \times n}; r_{ij} = \|p'_j - p'_i\|_2; \lambda$ 为平滑因子, λ 越大, $f(x, y)$ 越平滑; \mathbf{w} 为 $n \times 2$ 矩阵, $\mathbf{w} = (w_{ij})_{2 \times n}^T; \mathbf{a} = (a_{ij})_{2 \times 3}^T; \mathbf{b} = (f_1, f_2, \dots, f_n)^T, \mathbf{p} = (\tilde{p}'_1, \tilde{p}'_2, \dots, \tilde{p}'_n)^T$ 。 \tilde{p}'_i 为 p'_i 的齐次坐标,即 $\tilde{p}'_i = (x'_i, y'_i, 1)^T$ 。

如图 4 所示, $g(x, y)$ 和 $h(x, y)$ 表示 $f(x, y)$ 的两个分量。图 4(a) 绿色的点表示特征点映射偏差的 x 分量值, 图 4(b) 绿色的点表示特征点映射偏差的 y 分量值。通过对单应矩阵的结果进行弹性映射, 使得特征点 p_i 可以映射到 q_i , 进而引导整个图像的映射。相比全局单应矩阵映射, 经过弹性映射之后, 图像对齐更加准确。

可以根据形变函数 $g(x, y)$ 和单应矩阵 H 生成 I_2 在 I_1 坐标下的投影图 I'_2 (I'_2 也就是 I_2 的稳像图)。 I'_2 的生成方法如下。

对于任意像素 $p(x, y) \in I'_2$, 使用单应矩阵 H 得到映射点 $p'(x', y') = H[p(x, y)]$, p' 点的映射偏差为 $f(x', y')$, 消除映射偏差后, 对应 I_2 中的点为 $q = p' - f(p')$ 。使用双线性插值将 I_2 中 q 的像素值赋值给 $I'_2(x, y)$, 即

$$I'_2(p) = I_2\{H(p) - f[H(p)]\} \quad (29)$$

使用式(29)对 I'_2 中的所有像素赋值, 即可以得到 I_2 的稳像图 $I'_2(p)$ 。

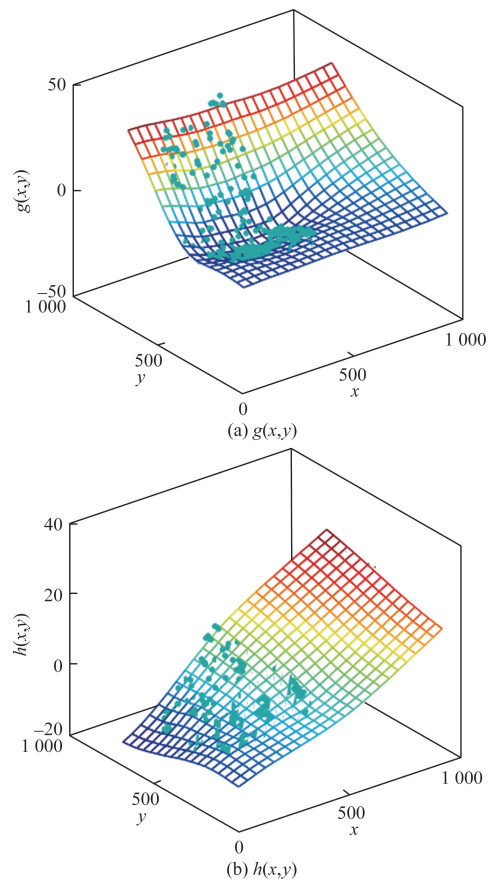


图 4 弹性映射法方法示意图

2 实验

采集铁塔上拍摄的视频作为实验视频。铁塔由于安装在室外,且高达数十米,没有其他物体阻

隔,更易受风等因素的影响产生抖动。且室外拍摄场景如海边、天空都属于弱纹理区域,更容易检验本文提出方法的效果。

姿态传感器的角速度数据频率为 200 Hz,视频帧率为 25 fps,当匹配到的特征点数量大于 1 000, RANSAC 方法内点数比率大于 0.8,且全局单应矩阵与单位矩阵的误差小于 0.001 时,重新将角速度数据归零,避免尺度漂移。

在特征点检测和匹配中,使用 FAST(加速段测试特征)+ORB(定向 FAST 和旋转二进制稳健独立基本特征)进行特征点检测和描述,并使用 KD (K 维)树进行特征点匹配,使用 RANSAC 方法通过计算全局单应矩阵获取式(13)中的置信度。匹配的特征点如图 5 所示,图 5 左上角给出了置信度结果。

为了证明使用姿态传感器紧耦合的方法比只使用图像估计全局单应矩阵的方法稳像效果更好,将间隔 1 s 的两帧稳像图像进行平均融合并对比融合图。即令 t 时刻的视频帧为 $I(t)$,令 $t+1$ 时刻的视频帧为 $I(t+1)$, $I(t)$ 和 $I(t+1)$ 进行映射后生成的稳像图像为 $I'(t)$ 和 $I'(t+1)$ 。那么融合图像为 $I_m = \frac{1}{2}[I'(t) + I'(t+1)]$ 。稳像效果越好, $I'(t)$ 和 $I'(t+1)$ 中非移动物体的图像位置越一致,图像越稳定,叠加效果越好。不然会存在较严重的

重影。图 6 和图 7 分别展示了使用图像计算全局单应矩阵和使用姿态传感器紧耦合方法生成的 I_m 。显然,相比纯图像方法,使用紧耦合方法的稳像效果更佳。

为了证明弹性映射方法的有效性,对比不加入和加入弹性映射生成的融合图像,如图 8 所示。由于风的影响,不仅相机会发生运动,场景中的树枝也会摆动。因此,全局的单应矩阵不能表征所有像素的映射。显然如图 8(a)所示,未添加弹性映射,树枝区域会存在重影,图像也更加模糊。添加弹性映射之后,融合图像更加清晰,无明显重影,表明了稳像效果确实得到了提升。

对比不同方法的速度。在 8 G 内存、2.4 GHz 主频的 PC 电脑上,对比全局单应矩阵方法、MeshFlow^[12]、Steadyflow^[13] 和本文方法在 1080P 视频上的运行速度,如图 9 所示。其中, Homography 表示基于图像的单应矩阵映射方法,其运行速度最快,只需要 0.025 s 就可以处理一帧图像。Sensor 表示本文提出的方法,本文方法加入了姿态传感器的角速度积分以及鲁棒弹性映射,比基于图像的单应矩阵映射方法稍慢,需要 0.036 s 完成一帧图像处理,但是仍然可以保证算法的实时运行。MeshFlow 需要计算全局单应矩阵,并对图像进行网格化,针对每个网格的光流还需进行两次中值滤波,因此需要



图 5 ORB 特征点匹配

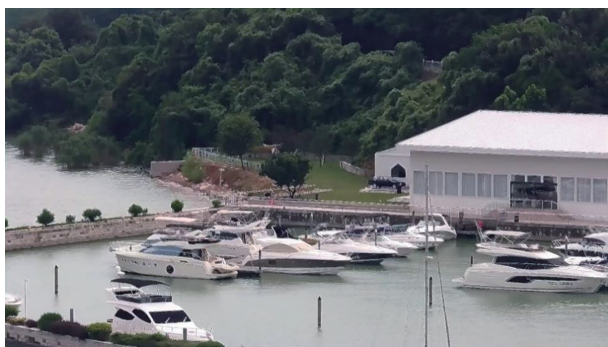


图 6 使用图像估计全局单应矩阵



图 7 使用姿态传感器紧耦合算法



(a) 未添加弹性映射



(b) 添加弹性映射

图 8 加入弹性映射前后对比

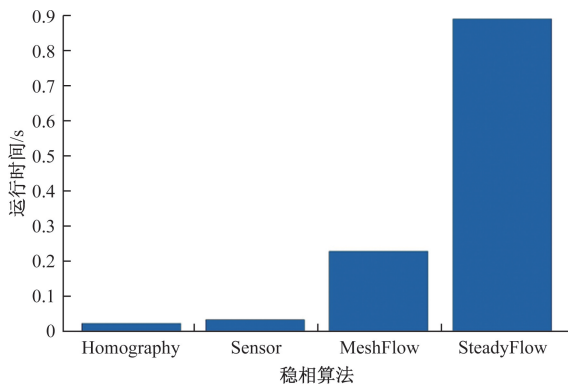


图 9 不同算法运行时间对比

0.3 s 左右完成一帧处理。最慢的是 SteadyFlow, 其需要计算整个图像的光流, 并进行前景提取和基于时间的滤波。综上所述, 本文算法在保证稳像视频质量的同时, 保证了算法的实时性。

3 结论

提出了一种基于姿态传感器紧耦合的视频稳像算法。通过对两帧图像间的姿态传感器角速度数据进行积分, 得到两个相机间旋转的初值。之后, 利用特征点的重投影误差以及最优旋转与姿态传感器计算的旋转之间的偏差对相机间旋转进行优化, 当特征点数量较少或者内点数较少时, 重投影误差的权重较小, 即最优旋转更加倾向于姿态传感器计算的旋转。相反, 计算可以使得重投影误差更小的旋转矩阵作为最优旋转矩阵。在全局单应

矩阵的基础上, 使用匹配的特征点引导图像的弹性映射, 使得匹配误差更小。通过实验表明, 与纯视频方法的稳像算法相比, 加入姿态传感器后, 稳像效果更佳。加入弹性映射方法之后, 局部的配准误差也可以进一步降低。同时, 与其他相同稳像质量的方法相比, 本文的方法运行效率也更高。

参考文献

- [1] 温鑫. 基于机场应用场景的视频监控系统设计与实现[J]. 科技和产业, 2020, 20(12): 237-241.
- [2] 唐晓亮, 朱瑞波, 朱从喜. 铁塔状态监测系统的构思与研究[J]. 科技和产业, 2018, 18(2): 119-122.
- [3] 魏闪闪, 谢巍, 贺志强. 新兴数字视频稳像中相机运动估计技术综述[J]. 计算机应用研究, 2017, 34(2): 321-327.
- [4] GUILLUY W, OUDRE L, BEGHDAI A. Video stabilization: overview, challenges and perspectives[J]. Signal Processing: Image Communication, 2021, 90: 116015.
- [5] ROBERTO E, SOUZA M, MAIA H A, et al. Survey on digital video stabilization: concepts, methods, and challenges[J]. ACM Computing Surveys (CSUR), 2022, 55(3): 1-37.
- [6] WANG Y, HUANG Q, JIANG C, et al. Video stabilization: a comprehensive survey [J]. Neurocomputing, 2023, 516: 205-230.
- [7] 李欣, 焦立男, 柳有权, 等. 一种基于改进 SIFT 的视频稳像方法[J]. 计算机与现代化, 2024(6): 43-50.
- [8] 王振宇, 王路倩, 王兴. 考虑相机运动补偿的视觉模式测试方法[J]. 振动工程学报, 2024, 37(6): 976-985.
- [9] 刘少波, 金恒宏, 钟德正. 基于平滑相机路径的电子稳像算法研究[J]. Operations Research and Fuzziology, 2024, 14: 143257.
- [10] LIU S, YUAN L, TAN P, et al. Bundled camera paths for video stabilization[J]. ACM Transactions on Graphics (TOG), 2013, 32(4): 1-10.
- [11] LIN K, JIANG N, LIU S, et al. Direct photometric alignment by mesh deformation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017: 2405-2413.
- [12] LIU S, TAN P, YUAN L, et al. Meshflow: minimum latency online video stabilization[C]//Computer Vision-ECCV 2016: 14th European Conference. Amsterdam: Springer International Publishing, 2016: 800-815.
- [13] LIU S, YUAN L, TAN P, et al. Steadyflow: spatially smooth optical flow for video stabilization[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014: 4209-4216.
- [14] CHOI J, PARK J, KWEON I S. Self-supervised real-time video stabilization[J]. arXiv preprint arXiv: 2111.2021: 05980.

- [15] SHI Z, SHI F, LAI W S, et al. Deep online fused video stabilization[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Santa Barbara; IEEE/CVF, 2022: 1250-1258.
- [16] ZHANG Z, LIU Z, ZENG B, et al. Minimum latency deep online video stabilization [J]. arXiv. 2022, 2212: 02073.
- [17] 张育铭, 王收军, 陈松贵. 基于IMU惯性传感器的防波堤护面块体位姿变化测量研究[J]. 科技和产业, 2022, 22(3): 292-296.
- [18] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. Orb-slam3: an accurate open-source library for visual, visual-inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.
- [19] LEE K, PARK H, KIM J. Geometric constraints for camera calibration without a calibration object in Proc[C]//IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). Vancouver; IEEE/CVF, 2023: 1234-1245.
- [20] LUO C, YANG W, HUANG P, et al. Overview of image matching based on ORB algorithm[J]. Journal of Physics: Conference Series, 2019(3): 032020.
- [21] SUN J M, SHEN Z, WANG Y, et al. LoFTR: Detector-free local feature matching with transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE/CVF, 2021: 8922-8931.
- [22] BROWN M, LOWE D G. Automatic panoramic image stitching using invariant features[J]. International Journal of Computer Vision, 2007, 74(1): 59-73.
- [23] LI J, WANG Z, LAI S, et al. Parallax-tolerant image stitching based on robust elastic warping [J]. IEEE Transactions on Multimedia, 2017, 20(7): 1672-1687.

A Video Stabilization Method Based on Tight Coupling with the Attitude Sensor

CHEN Shoufeng¹, LIU Yu², TANG Wujun¹, ZHANG Yifei¹, WANG Lehao¹

(1. China Tower Corporation Wuxi Branch, Wuxi 214111, Jiangsu, China;

2. China Tower Corporation Nanjing Science and Technology Innovation Center, Nanjing 211316, China)

Abstract: Since feature point matching and optical flow estimation are closely related to the image texture, a standalone visual approach for video stabilization may not be suitable for all scenarios was used. A tightly coupled attitude-sensor-based video stabilization method was proposed. By adaptively adjusting weights, the optimal homography between images relies more on the attitude sensor in low-texture areas or more on feature matching in rich-texture areas. After obtaining the optimal transformation, a robust elastic warping method was applied to further align consecutive image frames. Experimental results demonstrate that the proposed video stabilization method achieves better performance and robustness.

Keywords: attitude sensor; robust elastic warping; video stabilization