

NeuroSlam 视觉模块

1 概述

1.1 论文摘要¹

此方法不适用于 6 个自由度的无限制运动。

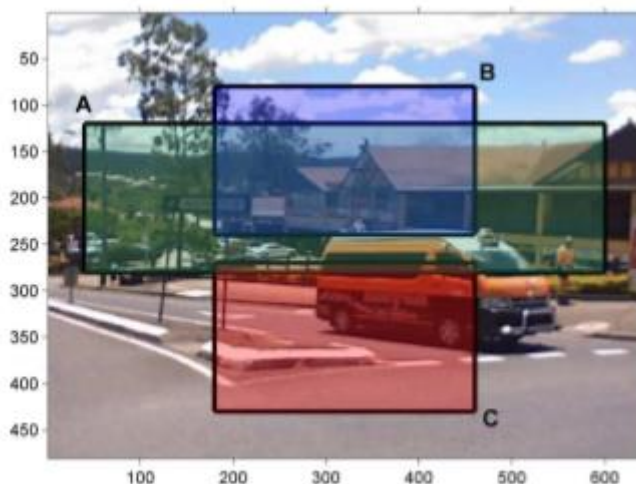
灰度影像降低分辨率以获得：航向角旋转速率、位移速率、高度变化速率和视觉模板匹配。

本文中**选择区域并裁剪**：将图像分为了 2 部分，分别用以计算航向角旋转速率、位移速率和高度变化速率。

NeuroSlam 的视觉部分主要应用了**区域分割和模板匹配**的方法。下面对区域分割和模板匹配作简要介绍。

1.2 区域分割²

在一幅分辨率为 480*640 的图像中，分割出三区域以用作不同目的。A 区域用作场景识别，B 区域用作旋转探测，C 区域包含丰富道路信息用作平移计算（原论文的场景是室外空间）。

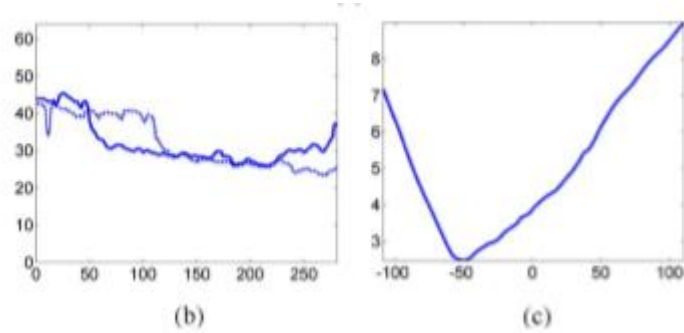


¹ NeuroSLAM: a brain-inspired SLAM system for 3D environments. Fangwen Yu. Jianga Shang

² Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System. Michael J. Milford, Member, IEEE, and Gordon F. Wyeth, Member, IEEE

1.3 模板匹配

将每个区域中的像素值按照列进行加和，形成 $1 \times N$ 的向量，称为扫描线。将扫描线用作匹配图片的模板。下图 b 是两幅影像的扫描线，c 图是两幅影像的像素偏移量。



下图是通过比较影像的扫描线来匹配的流程。存储前一帧的模板，与后一帧的扫描线进行比对，通过两者的差异来计算平移和旋转速率（原论文中将电脑固定在汽车顶部，俯仰角 pitch 为 0，忽略 pitch）。

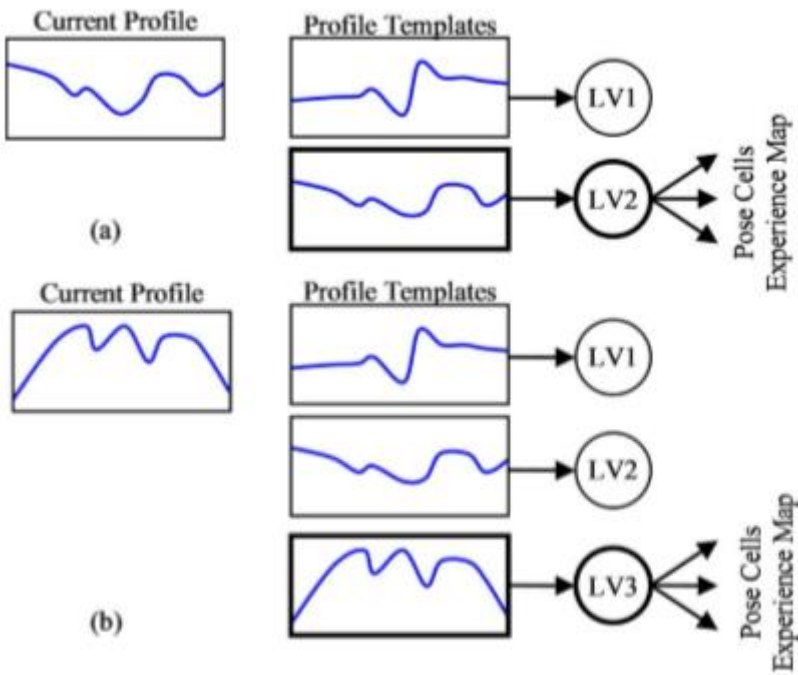


Fig. 7. Template matching system for local view calculation. (a) Where the current profile sufficiently matches a previously stored template, the associated local view cell is activated. (b) For a novel profile, a new template and local view cell is created and activated.

2 计算理论

2.1 扫描线和归一化

将图像每一列的像素进行加和，得到扫描线 I ($1*N$ 的向量)。

对扫描线像素进行归一化处理：

$$I'_{xy} = \frac{I_{xy} - \mu_{xy}}{\delta_{xy}},$$

where μ_{xy} is the mean. δ_{xy} is the standard deviation.

2.2 航向角旋转速率计算

航向角的旋转速率通过计算每列的像素最小偏移量来确定。 $i+1$ 帧和 i 帧相比偏移了 s^h 列，计算得到最小的偏移量为 s_m^h 。旋转速率=最小偏移量*经验参数。

I^i and I^{i+1} are shifted s^h in column dimension. Then, the average intensity difference between them is calculated.

$$d(I^i, I^{i+1}, s^h) = \frac{1}{w - |s^h|} \left(\sum_{n=1}^{w-|s^h|} \left| I_{n+\max(s^h, 0)}^{i+1} - I_{n-\min(s^h, 0)}^i \right| \right), \quad (27)$$

where s^h is the profile shift in column dimension. w is the width of the image.

$$s_m^h = \arg \min_{s \in [\rho^h - w, w - \rho^h]} d(I^i, I^{i+1}, s^h). \quad (28)$$

The rotational velocity $\Delta\theta$ is estimated by s_m^h multiplied by the constant σ^h . We can determine the σ^h empirically.

$$\Delta\theta = \sigma^h s_m^h. \quad (29)$$

2.3 估计平移速率

位移速率通过对比连续影像的之间的行差异，得到最小像素偏移量对应的最小像素强度信息差异值。

平移速率 = 最小像素强度信息差异值*经验参数(尺度)(平移速率需<阈值，否则维持上一帧不变)

$$v = \min[\mu d(I^i, I^{i+1}, s^h), v_{\max}]$$

μ 是经验值。

2.4 估计高程变化

高程通过减去行偏移后，计算列的像素偏移来确定。

The intensity difference $d()$ between I^i and I^{i+1} is calculated by

$$\begin{aligned} d(I^i, I^{i+1}, s_m^h, s^v) \\ = \frac{1}{h - |s^v|} \left(\sum_{m=1}^{h-|s^v|} \left| I_{m+\max(s^v, 0)}^{i+1} - I_{m-\min(s^v, 0)}^i \right| \right), \quad (31) \\ I = \sum_{j=1}^{w-|s_m^h|} I'_j, \end{aligned}$$

where s_m^h and s^v are offsets in column and row dimension, respectively. h is the height of the sub-image. w is the width of the sub-image. The d_m is the intensity difference in a set of consecutive images at the minimum offset.

$$\begin{aligned} d_m = \min_{\substack{s_m^h \in [\rho^h - w, w - \rho^h] \\ s^v \in [\rho^v - h, h - \rho^v]}} d(I^i, I^{i+1}, s_m^h, s^v), \quad (32) \\ v_h = \min[\mu d_m, v_{\max}^h], \end{aligned}$$

where the constant, μ , is calculated empirically for scaling physical speed.

3 代码实现

输入：序列图像

定义参数：分割区域的范围，旋转、水平平移和高度三个尺度，水平和竖直射场角，水平和竖直的像素平均偏移量，三个速率的阈值。

3.1 计算旋转和平移速率

- 1) 切割用以计算旋转和平移速率的图像区域，将图像每列加和，变成 $1 \times N$ 的向量，并归一化。
- 2) 对比当前向量与前一帧图像的向量，得到行向量最小像素偏移量和偏移量对应的两帧图像像素强度值差异。（计算参照公式 27、28）
- 3) 根据水平方向视场角计算得到每像素宽对应的单位视场角，旋转速率 = 单位视场角 * 最小像素偏移量 * 旋转尺度
- 4) 判断取值：旋转速率 > 旋转阈值时，旋转速率保持上一帧的值不变，否则更新旋转速率。
- 5) 平移速率 = 偏移量对应的两帧图像像素差异 * 平移尺度
- 6) 判断取值：平移速率 > 平移阈值时，平移速率保持上一帧的值不变，否则更新平移速率。

3.2 计算高程变化

- 1) 切割图像用以计算高程的部分，将图像每行加和，变成 $N \times 1$ 的向量。
- 2) 图像尺寸重定义为：水平方向偏移后。像素归一化，并计算每像素对应的竖直单位视场角。
- 3) 对比当前向量与前一帧图像的向量，得到列向量最小像素偏移量和偏移量对应的两帧图像像素值差异。（计算参照公式 31、32）
- 4) 高程变化速率 = 列向量对应的像素值差异 * 高程尺度
- 5) 判断取值：高程变化速率 > 高程变化阈值时，高程变化速率保持上一帧的值不变，否则更新高程变化速率。

3.3 轨迹变化

- 1) 旋转角度 = 初始角度 + 累加每帧旋转速率
- 2) $X = \text{初始轨迹} + \text{水平位移} * \cos(\text{旋转角度})$
- 3) $Y = \text{初始轨迹} + \text{水平位移} * \sin(\text{旋转角度})$
- 4) 高程 $H = \text{初始高程} + \text{累加每帧高程速率}$

4 总结

总结：该方法不同于传统视觉里程计依据特征匹配获取本质矩阵，也不同于

视觉陀螺仪通过灭点获取航向角，而是通过图像领域的模板匹配来计算角度和平移量，简化了计算，可用于实时制图与导航。但该方法同样使用了大量经验参数（尺度和阈值），稳定性尚待验证。

问题：模板匹配+视觉里程计的方法 2 篇原论文皆是应用于室外场景，迁移到室内场景，是否可行？